

UNIVERSIDADE FEDERAL DO PARANÁ

CLEVERSON SANTOS
LEONARDO GIOVANI SANDRINI
LUCAS BATISTA LATCHUK
MAURÍCIO MORAIS MARTINS

SISTEMA DE EVENTOS PARA A BICICLETARIA CULTURAL



CURITIBA
2016

UNIVERSIDADE FEDERAL DO PARANÁ

CLEVERSON SANTOS
LEONARDO GIOVANI SANDRINI
LUCAS BATISTA LATCHUK
MAURÍCIO MORAIS MARTINS

SISTEMA DE EVENTOS PARA A BICICLETARIA CULTURAL

Trabalho de Conclusão apresentado ao
curso de Tecnologia em Análise e
Desenvolvimento de Sistemas do Setor de
Educação Profissional e Tecnológica,
Universidade Federal do Paraná.

Orientadora: Prof.^a Rafaela Mantovani
Fontana

CURITIBA

2016

AGRADECIMENTOS

Agradecemos, primeiramente, à professora Rafaela Mantovani que aceitou nos orientar ao longo deste projeto, sendo sempre muito prestativa e atenciosa em todos os momentos que precisamos de seu apoio para desenvolvermos nossas atividades. Sem seu trabalho, a realização de nosso projeto como foi feito não seria possível.

Agradecemos também aos empresários Fernando Rosenbaum e Patrícia Valverde, que nos aceitaram nesta parceria, nos trazendo necessidades reais e possibilitando que tivéssemos um projeto mais “palpável” para trabalharmos.

Também gostaríamos de agradecer aos colegas da área que foram prestativos e nos apoiaram em momentos de dúvida com suas experiências e conhecimentos técnicos por vezes mais vastos que os nossos próprios.

Aos parentes e amigos, agradecemos pela compreensão em nossos momentos de ausência, em que nos dedicamos totalmente a este projeto, e pela união em nossos momentos de descanso.

Por fim, agradecemos à Universidade Federal do Paraná, por nos ceder este espaço e permitir que pudéssemos, dentro desta instituição, ampliar nossos horizontes, aprofundar nossos conhecimentos e evoluir, não só como profissionais, mas como pessoas.

SUMÁRIO

1	INTRODUÇÃO.....	4
1.1	Apresentação do tema	5
1.2	Objetivo Geral	6
1.3	Objetivos Específicos	6
1.4	Justificativa.....	7
1.5	Estrutura do documento	7
2	SISTEMAS DE INFORMAÇÃO	9
2.1	Base teórica	9
2.2	Comparativo com possíveis concorrentes	12
3	METODOLOGIA.....	14
3.1	Metodologia ágil	14
3.1.1	Sprint.....	15
3.1.2	Scrum.....	15
3.2	aplicação da metodologia.....	16
3.3	cronograma de atividades e responsabilidades	18
3.3.1	Sprint 0.....	19
3.3.2	Sprint 1	19
3.3.3	Sprint 2.....	21
3.3.4	Sprint 3.....	22
3.3.5	Sprint 4.....	22
3.3.6	Sprint 5.....	22
3.3.7	Sprint 5.....	23
3.3.8	Sprint 6.....	23
3.3.9	Sprint 7.....	23
3.3.10	Sprint 8.....	24
3.3.11	Sprint 9.....	24
3.3.12	Sprint 10.....	24

3.3.13	Sprint 11	25
3.3.14	Sprint 12	25
3.4	CONCEITOS DE UML	25
3.4.1	Diagrama de casos de uso	26
3.4.2	Diagrama de classes	26
3.4.3	Diagrama de sequência	26
3.5	Testes aplicados	27
4	APRESENTAÇÃO DO SISTEMA BICI CULTURAL EVENTOS	28
4.1	Aplicação web	29
4.2	Aplicação Mobile	44
5	CONCLUSÃO	51
5.1	Considerações quanto ao projeto	51
5.2	Ponderações quanto aos aspectos positivos e negativos	52
5.3	Considerações finais	52
6	REFERÊNCIAS	53
6.1	Apêndice A – Requisitos funcionais	53
6.2	Apêndice B - Requisitos não-funcionais	55
6.3	Apêndice C - diagramas de casos de uso	56
6.4	Apêndice D - diagrama de classes de análise	59
6.5	Apêndice E – Diagrama de Classes de implementação	61
6.6	Apêndice F – DIAGRAMA DE ATIVIDADES	63
6.7	Apêndice G – diagrama de sequência de registro de usuário	64
6.8	Apêndice H – diagrama de sequência do registro de eventos	65
6.9	Apêndice I – DIAGRAMA de sequência de busca por evento	67
6.10	Apêndice J – DIAGRAMA DE SEQUÊNCIA DE CADASTRO DE CURSO	68
6.11	APÊNDICE K – DIAGRAMA DE SEQUÊNCIA PARA POSTAGEM EM EVENTOS	69

6.12	Apêndice I – Diagrama de sequência para inscrição em evento	70
	Apêndice M – Descrição dos testes	71
	Requisito 1 (R1) – Formulários incompletos.....	71
	Requisito 2 (R2) – CPF inválido	72
	Requisito 3 (R3) – Campos numéricos.....	72
	Requisito 4 (R4) – Buscas	73
	Requisito 5 (R5) – Edição.....	73
	Requisito 6 (R6) – Exclusão	74

1 INTRODUÇÃO

No ano de 2015, de acordo com a UIT (União Internacional de Telecomunicações) o número de aparelhos celulares ultrapassou sete o da população mundial, de sete bilhões (EDGARD JÚNIOR, 2015). Com esse crescimento, novos *gadgets*¹ que se integram aos smartphones são desenvolvidos constantemente, de modo a facilitar atividades do cotidiano e, até mesmo, reimaginá-las, alterando nossas formas de percepção e interação.

Tais avanços levam muitos negócios, novos e tradicionais, a buscarem por meios de inovação com o uso de tecnologia, aplicando *hardwares* e *softwares* na otimização de atividades dos que produzem e de experiências para aqueles que já consomem seus produtos e/ou serviços, além de tentar alcançar novas audiências e engajar clientes em potencial.

Muito do que é produzido em *softwares* toma, senão como única plataforma, como foco principal o ambiente móvel. Isso se dá pelo crescimento do mercado de tecnologia mobile em todo o mundo. E no Brasil, isso não é diferente, tendo o quarto lugar nesse mercado, com mais de 272 milhões de linhas ativas (LEANDRA SOARES, 2014).

Outra tecnologia em constante ascensão é a web. Hoje, o Brasil já ocupa o 5º lugar do mercado (EXAME, 2014) garantindo muitos usuários online.

Levando tais dados em consideração, esse projeto teve como propósito implementar soluções em *web* e *mobile* de modo a otimizar processos em uma empresa com necessidades latentes por tecnologia.

Pensando nisso, se buscou elaborar um projeto de forma que pudesse explorar essa necessidade por tecnologia, ainda latente em muitas empresas que atuam em diversos ramos. Também, estabeleceu-se como proposta ir de encontro a ideais inovadores já trabalhados pelas empresas, de modo que fosse viável desenvolvê-los através das soluções em software planejadas.

Nesse sentido, fora encontrada a Bicletaria Cultural, empresa que presta diversos serviços relacionados a bicicletas, esses que vão além da manutenção,

¹ **Gadget:** é "pequeno utilitário desenvolvido para facilitar o acesso a funcionalidades disponibilizadas por determinadas aplicações mais abrangentes". **Fonte:** Infopédia

trabalhando com vendas, alugueis e, até mesmo, estacionamento. Também há, inerente às atividades da empresa, um forte incentivo pelo crescimento e disseminação de pensamentos e ideologias ciclo-ativistas, o que acaba por promover uma comunidade que entende o ciclismo como um bem social e um patrimônio cultural. Todos os processos da empresa são realizados com poucos recursos tecnológicos, contando apenas com planilhas eletrônicas para armazenar e organizar todos os dados, além do uso de papel em algumas atividades, o que vai contra os ideais ecológicos do empreendimento.

1.1 APRESENTAÇÃO DO TEMA

Os sócios proprietários da empresa Bicletaria Cultural atuam com ciclismo e ciclo ativismo há cinco anos em Curitiba. Nesse meio tempo, conquistaram diversos prêmios, dentre os quais foi escolhida como uma das quinze melhores ideias do mundo para coabitar espaços urbanos no concurso internacional *Smart Living Challenge*, na Suécia.

Como resultado desse reconhecimento pelas atividades desempenhadas na empresa, conquistaram grande visibilidade no meio ciclístico e respeito pelo conceito novo que estabeleceram.

No espaço da empresa, existe um ecossistema cultural que é muito maior do que a empreitada acerca das bicicletas. Outros empreendedores, com iniciativas que conversam com os ideais dos proprietários do local, também têm espaço para expor suas iniciativas, tornando o ambiente mercadologicamente plural, ainda que ideologicamente singular. Cursos sobre biodiversidade, exposições de plantas, leilões de arte, além da atuação por um estado de vigia política e apartidária realizada pelo “Movimento Primavera Cidadã” são algumas das ações que povoam o espaço compreendido pela Bicletaria Cultural.

Contudo, os proprietários sentem falta de um meio mais expressivo de apresentar a seus clientes todas as atividades propostas pela empresa e veem em recursos tecnológicos, dos quais não dispõem no momento, um meio para isso. Sua necessidade é por uma tecnologia que sirva tanto como meio de divulgação de todos os eventos realizados pela empresa, quanto como forma de

ingresso a esses eventos, realizando o pagamento através da solução desenvolvida.

Nesse sentido, os proprietários, clientes desse projeto, sofrem com a ausência de bons canais de comunicação entre eles e a comunidade que consome os produtos e serviços prestados pela empresa. Notificações de eventos, disseminações de notícias e informações ou mesmo ações de cunho cultural estão envolvidas na aplicabilidade dessa plataforma de comunicação em questão.

Eles necessitam de um canal que não só possibilite a interatividade com toda a comunidade que frequenta a Bicicletaria Cultural, como também a amplie em um sentido cicloativista, onde haja a conquista não somente de novos clientes mas de novos adeptos do ciclismo a partir da conscientização de tudo o que é promovido em função disso.

1.2 OBJETIVO GERAL

O projeto tem como propósito desenvolver um software *mobile* para a promoção e ingresso em eventos realizados pela Bicicletaria Cultural

1.3 OBJETIVOS ESPECÍFICOS

- Permitir que os administradores criem e gerenciem eventos através de uma plataforma *BackOffice* web.
- Possibilitar a compra de ingressos para os eventos através de um aplicativo móvel.
- Estreitar a comunicação entre os participantes dos eventos e seus promotores por meio de feedbacks dos eventos.
- Permitir que a aplicação seja mais um canal de informações relacionadas à Bicicletaria Cultural ao ciclismo.

1.4 JUSTIFICATIVA

Este projeto apresenta a criação de um aplicativo *mobile* onde os clientes da Bicicletaria Cultural terão acesso a todos os eventos realizados pela empresa, tendo a opção de compra de ingressos através do aplicativo.

O aplicativo móvel fora o meio pensado para isso, pois, segundo o IBGE (Instituto Brasileiro de Geografia e Estatística), o uso de *smartphones* é crescente, como mostra notícia de abril deste ano (G1, 2016), tendo se tornado o principal meio de acesso à web quando ultrapassou o uso de computadores em 2014, quando 80% da população brasileira já possuía *smartphones* e os usavam para acessar a internet, número superior aos 76,6% que, na época, realizavam acesso a rede por computadores.

Também é contemplado por este projeto o desenvolvimento de uma plataforma para administração dos eventos. Analisando vantagens e desvantagens entre aplicativos nativos e soluções web (MOBLEE, 2015), optou-se por desenvolver um sistema web responsivo por dar maior liberdade aos administradores, havendo a possibilidade de controlar os eventos tanto através de dispositivos móveis quanto por computadores pessoais.

1.5 ESTRUTURA DO DOCUMENTO

Este documento descreve todas as informações de relevância quanto ao desenvolvimento do projeto.

No capítulo 2, inicia-se uma contextualização quanto a sistemas de informação, descrevendo uma teoria para classificação de sistemas e apresentando todas as classes nela referidas. Com isso, buscou-se enquadrar o projeto aqui documentado nesta classificação, justificando porque este projeto pode ser compreendido em determinada classe. Após, é explanada a importância de desenvolver a aplicação descrita neste documento da forma como fora feita. Por fim, analisou-se outros aplicativos semelhantes já existentes no mercado, distinguindo-os quanto as suas funcionalidades.

Em seguida, o capítulo 3 contém informações referentes às metodologias utilizadas para gerenciamento e organização das atividades exercidas no

decorrer do desenvolvimento. Para cada tecnologia, há uma breve contextualização com o objetivo de apresentá-las sucintamente. Também neste capítulo, são descritos os passos realizados em cada etapa do desenvolvimento, discorrendo sobre o cronograma de forma mais detalhada.

No capítulo 4, são apresentadas as funcionalidades do sistema desenvolvido de forma instrutiva, com o propósito de capacitar o leitor deste documento a utilizar a aplicação. São exibidas as telas existentes no *software* e, em seguida, as funcionalidades apresentadas por cada uma delas individualmente.

Logo a seguir, no capítulo 5 procurou-se rever tudo o que fora realizado ao longo do projeto, analisando se o objetivo principal inicialmente proposto está de acordo com o resultado final entregue. Também, os objetivos específicos são avaliados para constatar se foram atingidos em conformidade com o planejado de início.

Finalmente, compilou-se os apêndices contendo os materiais desenvolvidos paralelamente, em apoio a documentação, como os diagramas e especificações UML. Além disso, listou-se as referências utilizadas para desenvolver esse documento, podendo ser consultada caso se faça necessário.

2 SISTEMAS DE INFORMAÇÃO

Comentado [RF1]: Vamos chamar de SISTEMAS DE INFORMAÇÃO

A utilização de sistemas de informação nos negócios vem se tornando comum mesmo em micro e pequenas empresas. Para a compreensão de sua relevância nesse cenário, cabe a explanação de suas definições, significados e classificações. Em seguida, apresenta-se os sistemas já existentes em comparativo com o *software* desenvolvido.

2.1 BASE TEÓRICA

Sistema de informação (SI) define-se como “um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informações” (LAUDON & LAUDON, 2011, p. 12).

As informações que transitam pelos SI referem-se a conjuntos de dados apresentados de maneira significativa para humanos. Em se tratando de negócios, as informações contêm dados sobre pessoas, locais e itens de relevância para a organização ou para o ambiente que as cercam. Já os dados contidos em cada informação, quando individualmente, são “sequências de fatos ainda não analisados, representativos de eventos que ocorrem nas organizações ou no ambiente físico, antes de terem sido organizados e arranjados de uma forma que as pessoas possam entendê-los e usá-los” (LAUDON & LAUDON, 2011, p. 12).

Dentro de uma organização, é possível que haja diferentes tipos de sistemas para atender a todos os interesses, especialidades e níveis presentes. A menos que se trate de uma suíte modular, um único sistema muito dificilmente proverá toda a informação de que um determinado negócio necessite.

De acordo com Laudon & Laudon (2006, p. 40), pode-se classificar sistemas de informação em seis grandes tipos. São eles:

- Sistemas de Suporte Executivo;
- Sistemas de Gestão de Informação;
- Sistemas de Suporte à Decisão;
- Sistemas de Escritório e;

- Sistemas de Processamento de Transações.

Cada um dos tipos citados aplica-se a um nível específico de operação dentro de uma organização.

Gerentes seniores utilizam Sistemas de Suporte Executivo como base para tomadas de decisão no nível estratégico da empresa. Esse tipo de sistema processa informações que requerem julgamento humano, pois referem-se a problemas nos quais é impossível estabelecer rotinas para que sejam solucionados. Essa classe de sistemas “é desenvolvida para incorporar dados sobre eventos externos como novas leis fiscais ou concorrentes, mas eles também desenham informação interna sumarizada” (LAUDON & LAUDON, 2006, p. 45). O trabalho desses sistemas é filtrar todos esses dados, deixando apenas o que é de alta criticidade de forma facilmente acessível e útil com pouco esforço e tempo. Para isso, costuma-se aplicar diversos recursos gráficos para reunir dados de múltiplas fontes e transformar a informação imediata para os executivos seniores.

Em um nível inferior de gestão, utiliza-se Sistemas de Suporte à Decisão. Com um propósito parecido com os sistemas da classe anteriormente citadas, esses diferem-se por apoiarem gerentes em decisões mais rápidas, com resultados em menor prazo. Os problemas tratados por sistemas desse tipo podem possuir rotinas que levem a solução, mas, em geral, não são bem especificados e precisam de atuação humana. Com grande poder analítico e que permite ao usuário inserir novos dados e mudar suposições, Sistemas de Suporte à Decisão são muito interativos e explicitamente amigáveis.

Abaixo, há os Sistemas de Gestão da Informação, que dão aos gerentes acesso a relatórios e têm a capacidade de manterem-se atualizados em tempo real com informações de performance da organização, além de armazenarem registros históricos. Geralmente, esses sistemas trabalham com informações internas e servem ao objetivo de estabelecer controle e planejamento dentro da empresa através de relatórios operacionais.

Já os Sistemas de Trabalho de Conhecimento e de Escritório atendem ao nível de conhecimento da organização, sendo utilizados por trabalhadores de conhecimento e trabalhadores de dados respectivamente. Os trabalhadores do conhecimento são definidos como “pessoas que possuem graduação universitária formal e que são frequentemente membros de profissões

reconhecidas como engenheiros, doutores, advogados e cientistas" (2006, p. 43). Seu trabalho é o de "criar nova informação e conhecimento" (LAUDON & LAUDON, 2006, p. 43). Já os trabalhadores de dados são descritos como tendo "graduação menos formal de educação avançada e tendem a processar mais do que criar informação" (LAUDON & LAUDON, 2006, p. 43). Os sistemas usados por esses profissionais têm o propósito de apoiar e coordenar as atividades desenvolvidas no escritório.

Finalmente, há os Sistemas de Processamento de Transações, que são os sistemas mais elementares de uma organização e processam dados em nível operacional. Estes sistemas executam transações diárias e registram rotinas para a manutenção das operações essenciais ao negócio, como vendas, pagamentos, reservas e transferências. Nesse nível não há espaço para análises, todos os processos são pré-definidos e bem estruturados.

Sistemas dessa classe podem ser subdivididos em cinco áreas de aplicações dentro da organização:

- Vendas e Marketing;
- Manufatura e produção;
- Finanças e contabilidade e;
- Sistemas únicos para indústrias específicas.

Segundo a definição de Sippu & Soisalon-Soininen (2015, p. 1), "transação é uma sequência de ações de leitura e atualização no banco de dados, realizadas no banco sob uma sequência de requisições da aplicação".

A aplicação desenvolvida pode ser classificada como um Sistema de Processamento de Transações (SPT), pois este projeto implementa as operações de processo de pagamentos para compra de *tickets* que dão acesso aos eventos, e anúncio e divulgação de novos eventos.

Sistemas de Processamento de Transações têm aplicações comerciais, trabalhando em informações detalhadas para a atualização de registros em banco. Esta classe de sistema é utilizada para dar suporte a uma ou mais atividades relevantes ao negócio.

2.2 COMPARATIVO COM POSSÍVEIS CONCORRENTES

Quanto às funcionalidades, o aplicativo a ser desenvolvido enquadra-se no nicho de guias de eventos. Em relação a esse tipo de aplicações, existem diversas opções já desenvolvidas e, para que fosse desenvolvido um software interessante para o cliente, foram implementadas funcionalidades comuns a outros aplicativos, acrescentando-se as necessidades específicas do cliente.

Realizou-se a análise de X aplicativos semelhantes ao proposto neste trabalho. A Tabela X apresenta a lista de funcionalidades verificadas e aponta as encontradas em cada aplicativo.

Começando pelas funcionalidades de compra através do aplicativo, de criação de tipos diferentes de ingresso e da divulgação desses eventos em redes sociais, são funcionalidades comuns a quase todos os concorrentes já existentes (“Eventbrite”, “Vamos”, “All Events in City” e “Aplicativo Ticket”) a exceção de um (“Nearify”), embora duas destas aplicações utilizem uma API do “Eventbrite” para realizar essas atividades (“Vamos” e “All Events in City”).

Também, a busca por eventos e a visualização deles organizada por tipo são abrangidas em quase todos os guias de eventos do mercado, a exceção do aplicativo “Vamos”.

A opções começam a afunilar quanto a algumas funções como a possibilidade de comentar em eventos, presentes apenas nos aplicativos “Nearify” e “AppTicket” (o aplicativo “All Events in City” permite comentários através de uma API do Facebook).

Outras funcionalidades só são vistas no “Nearify”, como a visualização de eventos por data e a possibilidade de avaliação através do aplicativo.

Contudo, uma função esperada pelo cliente e que não é vista em qualquer outro aplicativo é um espaço para feedback mais aprofundado, que permita o envio de opiniões escritas e fotos dos eventos tiradas por quem participou. Essa funcionalidade amplia o poder da aplicação enquanto ferramenta estatística para que o cliente tenha parâmetros do que pode ser melhorado, aproveitado ou descontinuado em eventos futuros.

TABELA 1 – QUADRO-RESUMO DE COMPARAÇÃO COM COMCORRENTES

Aplicativo ->	Nosso App	Eventbrite	Nearify	Vamos	AppTicket(Ionic)	All Events in City
Compra de eventos no App	X	X		X(Via Eventbrite)	X	X(Via Eventbrite)
Tipos de ingresso	X	X		X(Via Eventbrite)	X	X(Via Eventbrite)
Eventos por tipo	X	X	X	X(Via Eventbrite)	X	X(Via Eventbrite)
Busca de eventos por localização		X	X	X		X
Eventos por TAG		X	X	X		
Eventos por data	X		X			
Eventos por preço		X	X			
Adicionar ao calendário			X			X
Mapa com localização do evento			X	X		X
Compartilhar eventos nas mídias soc	X		X	X	X	X
Comentário com texto	X		X		X	
Comentário com fotos	X		X		X	X(Via Facebook)
Pessoas com mesmo gosto artístico			X			
Eventos por artista			X			
Histórico de eventos compartilhados			X			
Histórico de atividades			X	X	X	X
Ativar e desativar notificações	X		X			X
Favoritar um evento			X	X	X	
Avaliar um evento	X		X			
Atividade após ocorrência de evento	X					
Pesquisa de eventos	X	X	X		X	X
Editar Perfil	X	X	X		X	X
Escolher áreas de interesse		X	X			X
Código promocional		X		X(Via Eventbrite)		X(Via Eventbrite)

Legenda: Quadro-Resumo de funcionalidades presentes no sistema desenvolvido nesse projeto em comparativo com os aplicativos já presentes no mercado. Fonte: Dados adaptados da *Play Store*. Fonte: Dados adaptados de *PLAY STORE* (2016).

Baseando-se em pesquisas sobre classificação de sistemas, conclui-se que este projeto está definido como um SPT (Sistema de Processamento de Transações) que viabiliza a promoção de eventos realizados pela Bicletaria Cultural.

Em comparação com seus concorrentes, sua diferença mais notável é o contato pós-evento que é realizado através do aplicativo *mobile* possibilitando que os promotores tenham uma visão de como sua proposta foi recebida pelo público.

A seguir, é apresentado como este projeto organiza-se tecnologicamente e este documento se aprofundará nas ferramentas utilizadas gerencialmente para assegurar que tudo o que fora proposto pudesse ser entregue em seu prazo final sem qualquer falha técnica.

3 METODOLOGIA

Ao longo deste projeto, fora realizada uma pesquisa para levantar possíveis caminhos metodológicos a serem seguidos a fim de alcançar um resultado final com entrega de qualidade.

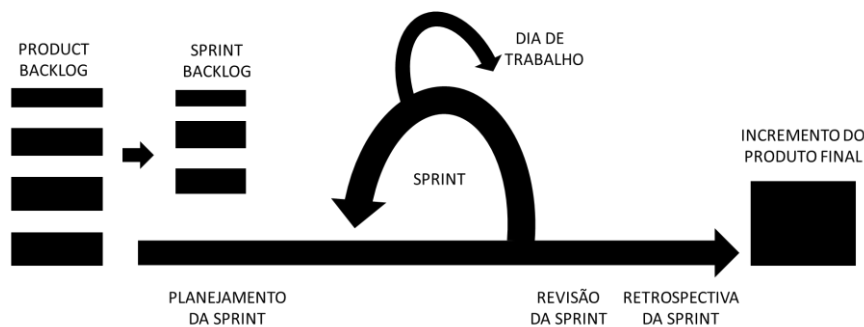
Desta forma, optou-se por utilizar métodos já estabelecidos no mercado como a metodologia ágil quanto a gestão, que é o primeiro tópico apresentado nesse capítulo.

3.1 METODOLOGIA ÁGIL

Métodos ágeis de desenvolvimento de software tomam como base filosófica o Manifesto Ágil, criado em fevereiro de 2001 por um grupo de profissionais que trabalhavam com desenvolvimento de software e buscavam alternativas às maçantes metodologias tradicionais. Neste manifesto é declarada a valorização de “Indivíduos e a interação entre eles mais que processos e ferramentas; *software* em funcionamento mais que documentação abrangente; colaboração com o cliente mais que negociação contratual; responder a mudanças mais que seguir um plano” (BECK, et al.,2001).

Dentre os métodos existentes, utilizou-se a metodologia *Scrum*, “um *framework* Ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos” (SABBAGH, 2013, p.17), separando as etapas de desenvolvimento em *sprints*, exemplificado na Figura 1, com período de duração semanal.

FIGURA 1 – PROCESSO DE UMA SPRINT



Legenda: Exemplo gráfico de um ciclo de atividades dentro de um projeto Ágil. Adaptado de DEVMEDIA, 2015.

3.1.1 Sprint

Sprint é o nome dado à um ciclo de desenvolvimento iterativo e incremental dentro de um projeto, onde determinadas funções são desenvolvidas para serem entregues de forma executável em um pacote ao fim deste ciclo.

“Iterativo” e “incremental” são bases que definem o desenvolvimento ágil, incluindo-se neste o *scrum*.

Por “incremental”, entende-se que o sistema é desenvolvido de modo fragmentado, pedaço a pedaço, onde são construídas as funcionalidades mais básicas de início, até finalizar com as implementações mais complementares.

Já o uso do termo “iterativo” dá-se por conta da ideia de ser quase impossível desenvolver uma funcionalidade completamente perfeita de início. Iteração é esse “retrabalho” que se tem ao aperfeiçoar uma funcionalidade.

3.1.2 Scrum

Por *Scrum*, entende-se “um *framework* iterativo e incremental para desenvolver produtos” (MEDEIROS, 2013, p.7). Sua concepção data dos anos 1990, mas sua popularização deu-se na década seguinte. Atualmente, sua utilização em projetos de desenvolvimento de software supera a de metodologias

tradicionais. Em pesquisa realizada em 2011, Instituto *VersionOne*, Scrum fora apontado como método de trabalho utilizado por dois em cada três participantes (SABBAGH, 2013, p.25).

Aplicado a este projeto, possibilitou diversos ganhos, o dinamismo para encontrar a melhor forma de organização dos processos de desenvolvimento, dando visibilidade das práticas que estão funcionando bem e das que precisaram ser revistas ou descontinuadas.

Também, a constante entrega de pacotes executáveis apresenta valor mais imediato ao cliente, além de permitir mudanças quase que imediatas, se necessário, causando pouco impacto ao *deadline* do projeto e tornando-o de baixo risco para os desenvolvedores.

Além disso, a linha de aprendizado progressiva ao longo do projeto permite que mais entregas sejam realizadas em menor tempo otimizando o tempo de desenvolvimento. Esta evolução torna-se visível pelo que já fora implementado até então.

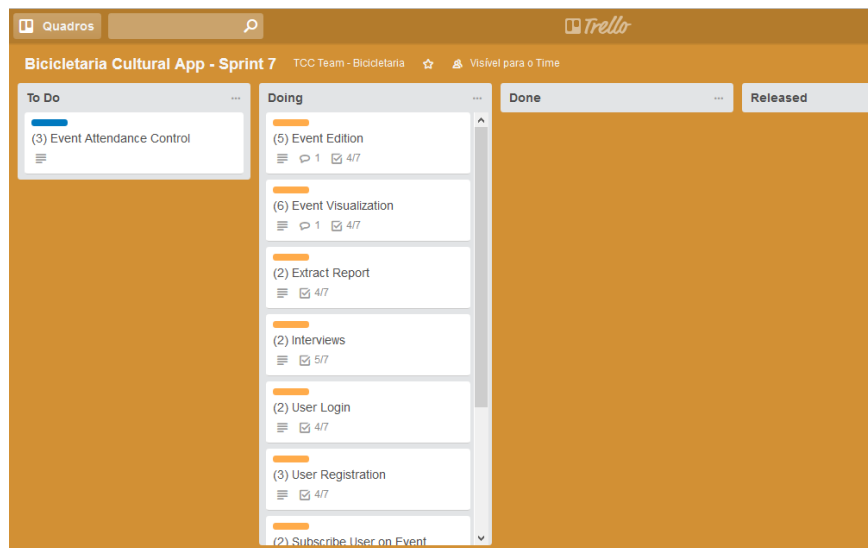
3.2 APLICAÇÃO DA METODOLOGIA

Uma das características mais presentes na aplicação da metodologia *Scrum* em projetos é a realização de reuniões diárias. Para que todos os integrantes da equipe pudessem manter suas rotinas diárias sem prejudicar o processo de desenvolvimento, a grande maioria das reuniões foi realizada remotamente, utilizando softwares de conversa *online* como, por exemplo, o *Skype* como meio de comunicação. Esta ferramenta permite o compartilhamento da tela, o que foi útil para que todos pudessem compartilhar suas atividades em tempo real. As demais reuniões ocorreram pontualmente, conforme necessário, dentro da Universidade Federal do Paraná, em sua grande maioria com presença da professora orientadora deste projeto.

Outra característica de *Scrum* incorporada a este projeto é o desenvolvimento através de sprints. Para controle da própria equipe, utilizou-se a ferramenta *online Trello*, que permite o acompanhamento progressivo das atividades a partir de quadros criados pelos envolvidos, mostrando o que foi feito

em cada etapa por cada um e em que status cada atividade se encontra como exemplificado na Figura 2.

FIGURA 2 – EXEMPLO DE ATIVIDADES DE UMA SPRINT CONTROLADAS ATRAVÉS DA FERRAMENTA *TRELLO*



Legenda: *Backlog* de tarefas é representado pela caixa mais à esquerda contendo o que ainda deve ser feito, é apresentado tudo o que está sendo feito na Sprint em questão (Sprint 7, neste caso). As duas últimas caixas recebem o que já foi feito, no caso da primeira, e o que já foi entregue ao cliente, no caso da segunda. Fonte: Os Autores.

Ao início do projeto, as atividades foram divididas conforme a disponibilidade e conhecimento técnico de cada integrante para realiza-las. Assim, o acadêmico Cleverson Santos, ficou responsável pelo desenvolvimento do *back-end* das aplicações *web* e *mobile*, Maurício Morais e Lucas Latchuk atuaram no desenvolvimento do *front-end* de ambas as aplicações e Leonardo Sandrini responsabilizou-se por documentar todo o projeto.

Contudo, dificuldades encontradas pelos integrantes responsáveis por desenvolver o sistema acarretaram em mudanças na organização da equipe e na divisão de trabalho. Ao fim deste projeto, Maurício Morais seguiu atuando unicamente no desenvolvimento do *front-end web*, enquanto Lucas Latchuk assumiu as atividades referentes ao *front-end mobile*.

3.3 CRONOGRAMA DE ATIVIDADES E RESPONSABILIDADES

Ao iniciar-se este projeto, todas as atividades necessárias para a sua concretização foram listadas e seu tempo para desenvolvimento foi estimado.

Abaixo, está a tabela 2, que apresenta o cronograma de trabalho planejado de início. Durante o processo de desenvolvimento, algumas datas foram postergadas por problemas encontrados pontualmente e outras foram antecipadas, graças à velocidade de produção ascendente dos desenvolvedores.

TABELA 2 – CRONOGRAMA DE ATIVIDADES RELATIVAS AO PROJETO

Nº da Sprint	Data de Início	Metas da Sprint
0	03/08/2016	Elaborar as histórias de usuário; Ler o capítulo 13 do livro "Desenvolvimento de Software com Scrum; Elaborar o planejamento de releases; Capítulo 1 da Documentação.
1	12/08/2016	<i>User Administrator Login;</i> <i>User Administrator Registration;</i> <i>Event Registration.</i>
2	19/08/2016	<i>User Registration;</i> <i>User Login;</i>
3	26/08/2016	<i>Event Visualization;</i> <i>Event Edition;</i> Capítulo 2 da Documentação.
4	09/09/2016	<i>Search Event;</i> <i>Event Post;</i> <i>Share Event On Social Media.</i>
5	16/09/2016	<i>User Profile;</i> <i>User History;</i> Capítulo 3 da Documentação.
6	30/09/2016	<i>Event Rating;</i> <i>App notification;</i> <i>Email Notification.</i>
7	07/10/2016	<i>Subscription Details;</i> <i>Manage User Post;</i> Capítulo 4 da Documentação.
8	21/10/2016	<i>Subscribe User on event;</i> <i>User Registration on Event;</i>

<i>Interviews.</i>		
9	28/10/2016	<i>Event Attendance Control;</i> <i>Extract Report;</i> Capítulo 5 da Documentação.
10	11/11/2016	Integração do sistema; Revisão da Documentação.
11	18/11/2016	Entrega do sistema; Preparação da apresentação.
12	08/12/2016	Apresentação para a banca; Liberação do Sistema (Go Live).

Legenda: As atividades são apresentadas na forma cronológica em que foram planejadas. Fonte: Os autores.

3.3.1 *Sprint 0*

Ao apresentar a proposta de desenvolvimento para Trabalho de Conclusão de Curso para a Professora Orientadora Rafaela Mantovani, elaborou-se o plano de *releases*², onde as atividades a serem feitas foram todas levantadas e estimadas em tempo de desenvolvimento.

Além disso, foi solicitado que o capítulo 1 da documentação tivesse uma primeira versão escrita.

3.3.2 *Sprint 1*

Foi realizado um diagrama de casos de uso e um diagrama de classes para que a equipe tivesse uma visão prévia do que seria preciso implementar. Uma primeira versão do capítulo 1 foi entregue, analisado de pronto pela professora orientadora e devolvido para revisão.

O ambiente de desenvolvimento foi configurado na máquina de todos e as tecnologias necessárias para o trabalho foram definidas e preparadas. Assim, o desenvolvimento foi iniciado.

² **Release:** “Release é a entrega de um ou mais Incrementos do Produto prontos, gerados pelo time de desenvolvimento em um ou mais *Sprints* sucessivos”. Fonte: *Scrum – Gestão Ágil para projetos de sucesso*.

3.3.2.1 Tecnologias aplicadas

Este projeto buscou utilizar-se de diversos recursos de desenvolvimento disponíveis para obter aplicações concisas e estáveis para atender às necessidades do cliente de forma satisfatória dentro do prazo previamente estipulado. As escolhas feitas do que seria utilizado deu-se da procura por tecnologias gratuitas que auxiliassem no desenvolvimento de software com qualidade. A tabela 3 mostra um quadro-resumo das tecnologias aplicadas ao projeto.

TABELA 3 – QUADRO-RESUMO DE TECNOLOGIAS UTILIZADAS NO PROJETO

Tecnologia	Versão	Aplicação
Ionic	v1.3.2	Desenvolvimento mobile
AngularJS	1.5.8	Desenvolvimento front-end
Hibernate	5.2.4.Final	Persistência no banco de dados
Spring MVC	1.4.2	Gerenciamento da aplicação
Spring Boot	1.4.2	Inicialização sem deploy

Fonte: Os autores.

Primeiramente, pode-se citar a utilização de *Ionic*, um *framework Javascript* que permite a criação de aplicações *cross-mobile*, ou seja, executáveis em diversos sistemas operacionais móveis, de maneira mais rápida e simplificada. Este *framework* “oferece um ecossistema de desenvolvimento completo e de alta performance” (HARTINGTON, 2015, p.7), sendo, para este projeto, uma excelente alternativa para o desenvolvimento nativo.

Seu potencial de performance por se basear fortemente em tecnologias que o permitem implementar grande parte de suas funções no lado do cliente, enquanto o servidor se preocupa apenas em quais telas deve apresentar.

O *Ionic* é baseado em outra tecnologia *Javascript* chamada *AngularJS*. Esta última, também aplicada ao projeto, é um *framework MVC (Model-View-Controller)* aplicada no lado do cliente em conjunto com outras tecnologias *front-end*: HTML, CSS e *Javascript*.

Segundo Flávio Almeida (2015) o foco de *AngularJS* é a SPA (*Single Page Application*), modelo de desenvolvimento em que a maior parte da

codificação é feita no *front-end*. Almeida descreve *AngularJS* como “um *high opinionated framework*, o que significa que ele guia o desenvolvedor de maneira implacável na organização do seu código” (2015, p.52).

O criador deste framework foi Misko Hevery, que buscou aplicar no cliente as vantagens presentes no MVC existente no servidor, como os componentes reutilizáveis, a possibilidade de realizar testes e a fácil manutenção.

Todas as tecnologias citadas permitem que os softwares desenvolvidos neste projeto sejam performáticos e dinâmicos, além de auxiliarem no processo de desenvolvimento com recursos complementares, inexistentes na linguagem sob as quais são construídas.

No lado do *back-end*, implementado em *Java*, linguagem programação orientada a objetos, também há interação com outras tecnologias que apoiam o desenvolvimento.

A começar pelo Hibernate, um *framework* de JPA (Java Persistence API). Em outras palavras, esse é um recurso que auxilia a aplicação a persistir no banco de dados. Com ele, o *schema* do banco de dados pode ser facilmente exportado. Também, possui recursos para realizar engenharia reversa no banco e *annotations* que simplificam a utilização do *framework*.

Junto com Hibernate, esse projeto utiliza *Spring MVC*, que auxilia no gerenciamento da aplicação com recursos como a criação de lógicas através de arquivos *XML* ou de *annotations*, por exemplo. Seu trabalho é tornar a infraestrutura do código mais flexível e dinâmica, facilitando, por exemplo, modificações sem causar impacto em outros trechos de código. O propósito é que o desenvolvedor possa focar nas regras de negócio e não perca tempo estruturando as classes de maneira eficiente.

Juntamente com o *Spring MVC*, aplicou-se a este projeto o *Spring Boot* que facilita a execução da aplicação sem a necessidade de *deploy* em servidor.

3.3.3 Sprint 2

Houve pequenas mudanças nas tecnologias aplicadas por problemas enfrentados com as escolhas iniciais, substituindo o Bootstrap convencional por

uma versão alternativa. O *login* de administrador foi finalizado e um diagrama de atividades também foi entregue. Uma nova versão do capítulo 1 foi entregue para análise com correções quanto à extensão e organização de parágrafos e pessoa na qual o texto estava discursado. Após esta, foi devolvido para nova revisão.

3.3.4 *Sprint 3*

Dando continuidade ao processo de desenvolvimento, nessa *Sprint* a equipe buscou finalizar os demais componentes que correspondem a classe de administrador tanto no *back-end* quanto no *front-end* da versão web do sistema.

Uma nova versão do capítulo 1, com correções quanto a formatação e a informações sem dados estatísticos, e uma primeira do capítulo 2, conceituando teoricamente o sistema desenvolvido por este projeto, foram entregues para análise da professora orientadora e devolvidos para revisão após serem analisados.

3.3.5 *Sprint 4*

Nesta *Sprint*, são finalizados os componentes de *back-end* do administrador e iniciados os de usuário. Problemas com a tecnologia escolhida para auxiliar no estilo do *front-end* atrasaram a finalização dos componentes do administrador.

Também fora apresentada uma nova versão da documentação, com um capítulo 2 mais amplo, com mais conteúdo, tomando livros técnicos para embasar as informações descritas.

3.3.6 *Sprint 5*

Nesta *Sprint*, foram finalizados os componentes de *back-end* do administrador e iniciados os de usuário. Problemas com a tecnologia escolhida para auxiliar no estilo do *front-end* atrasaram a finalização dos componentes do

administrador. A tecnologia foi substituída para evitar que o desenvolvimento fosse prejudicado.

Quanto a documentação, uma nova versão dos capítulos 1 e 2 foi enviada para análise, com várias alterações quanto a referências, forma como o texto foi escrito e informações a se acrescentar.

3.3.7 *Sprint 5*

Com os componentes de administrador já finalizados, a equipe começou a trabalhar na parte de eventos presente no sistema web tanto no *front* quanto no *back-end*.

Foi feito um primeiro modelo de um diagrama de sequência é elaborado.

3.3.8 *Sprint 6*

Dando continuidade aos componentes de eventos, nessa *Sprint* continuaram sendo desenvolvidas as telas e o back-end que, a pedido do cliente, deveriam distinguir cursos de outros tipos de eventos.

Uma nova versão do diagrama de sequência fora entregue para ser analisada, assim como os capítulos 1 e 2 novamente revisados para corrigir pendências e uma primeira versão do capítulo 3, contendo informações sobre a metodologia e as tecnologias aplicadas ao projeto.

3.3.9 *Sprint 7*

Nesta *Sprint*, o acadêmico Lucas Latchuk deixa de apoiar Maurício Morais no desenvolvimento do *front-end* web e passa a atuar sozinho desenvolvendo o *front-end* mobile com uso da tecnologia *ionic*.

Ao alterar o modo de divisão das atividades, a produção do projeto avança, tendo vários componentes sendo desenvolvidos individual e paralelamente.

A documentação ganhou uma nova versão do capítulo 3, ampliando as informações quando ao trabalho desempenhado ao longo do projeto.

3.3.10 *Sprint 8*

Na *Sprint 8*, o sistema já estava bem avançado tanto na parte web quanto na mobile com muitas telas finalizadas. Últimos componentes de eventos foram implementados como a geração de relatórios e o controle de frequência.

Na documentação, o capítulo 4 com uma apresentação concreta do sistema é escrito. Nele havia informações de como utilizar o sistema. Os diagramas de sequência foram finalizados.

3.3.11 *Sprint 9*

Nesta *Sprint*, o sistema, até então desenvolvido de forma fragmentada por cada um dos integrantes, passou a ser integrado para funcionar como uma única aplicação.

O capítulo 5 foi escrito contendo uma conclusão para o projeto, analisando se o objetivo foi concluído e quais foram os pontos positivos e negativos encontrados ao longo do desenvolvimento.

3.3.12 *Sprint 10*

Nesta *Sprint*, segue o trabalho de integração do código desenvolvido no *back-end* com as telas que foram desenvolvidas. O sistema começou a passar por testes.

O documento é passa por revisão antes de ser entregue a professora orientadora.

3.3.13 *Sprint 11*

O sistema passou por correções de bugs após os testes para ser entregue sem erros antes da apresentação para a banca.

Uma apresentação do projeto, em powerpoint, foi preparada para o dia da defesa.

3.3.14 *Sprint 12*

Essa *Sprint* ocorre após a entrega do projeto, quando o sistema é apresentado e defendido para a banca.

Com propósito o acadêmico de graduação no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, este é o fim do projeto. Contudo, por contar com clientes reais, os softwares desenvolvidos nesse projeto passarão a auxiliar nas atividades de organização de cursos e eventos na Bicletaria Cultural.

No próximo capítulo, são apresentadas as funcionalidades de cada aplicação desenvolvida de modo a instruir os usuários em sua utilização.

3.4 CONCEITOS DE UML

A modelagem do sistema desenvolvido neste projeto deu-se através da utilização de diagramas que seguem o padrão UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada), que Fowler (2005, p.25) define como “uma família de notações gráficas apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de *software*, particularmente daqueles construídos utilizando o estilo orientado a objetos (OO) ”.

Alguns dos artefatos produzidos ao longo deste projeto, partindo desta linguagem gráfica, são conceituados a seguir.

3.4.1 Diagrama de casos de uso

Os diagramas de casos de uso são utilizados para “captar os requisitos funcionais de um sistema” (FOWLER, 2005, p. 104). Nele, são apresentadas as interações entre os possíveis usuários e o sistema de modo a apresentar como o sistema é utilizado numa narrativa gráfica.

Analisando um determinado cenário, onde o ator, que representa um usuário, deverá seguir uma série de passos para realizar uma ação dentro do sistema, pode-se extrair um ou mais casos de uso, de acordo com a complexidade deste cenário, como apresentado pela Figura 31 (p.60).

3.4.2 Diagrama de classes

Já os diagramas de classes, os mais amplamente usados segundo Fowler (2005), descrevem “os vários tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos entre eles” (FOWLER, 2005, p. 52).

Também é seu propósito apresentar as propriedades e as operações de uma classe, por padrão chamando ambas de características. As características de estrutura de uma classe estão compreendidas nas propriedades, como se correspondessem aos campos dela a exemplo da Figura 34 (p. 65).

3.4.3 Diagrama de sequência

Segundo Fowler (2005) a função do diagrama de sequência é capturar um único comportamento dentro de um cenário e mostrar os presentes objetos e as mensagens que são passadas entre eles, como é mostrado a partir da Figura 36 (p.68). Para ele “os diagramas de sequência mostram uma interação, exibindo cada participante como uma linha de vida, que corre verticalmente na página, e a ordem das mensagens, lendo a página de cima para baixo” (FOWLER, 2005, p. 67).

3.5 TESTES APLICADOS

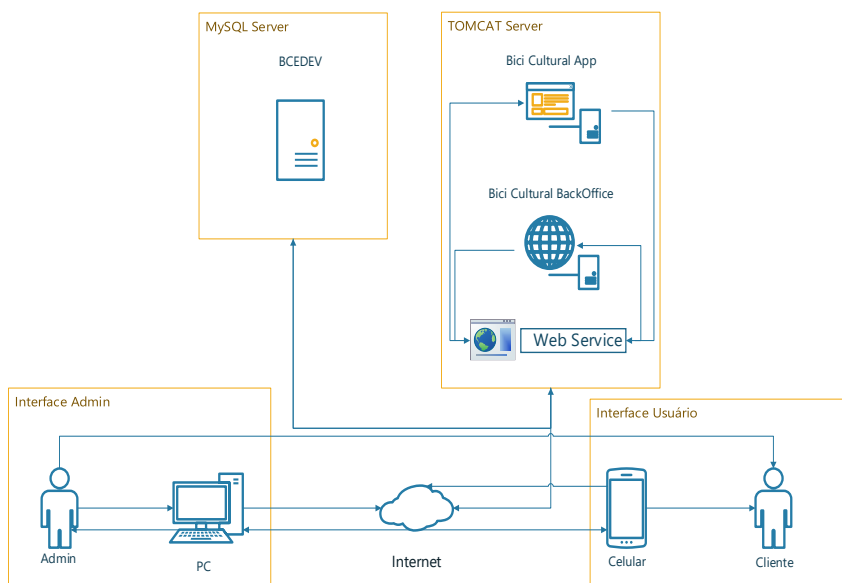
A realização de testes é feita com o objetivo de “mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso” (SOMMERVILLE, p.144).

Foram realizados testes caixa-preta, referentes à interface do software. Ou seja, foram testadas as funcionalidades, sem avaliar a estrutura dos códigos executados. A descrição dos testes encontrasse no Apêndice M ().

4 APRESENTAÇÃO DO SISTEMA BICI CULTURAL EVENTOS

Como já citado anteriormente, o sistema desenvolvido conta com dois softwares que se comunicam. Um deles, uma aplicação web que é executada no servidor e é acessada por nossos clientes para que possam gerenciar todo o sistema. O segundo, um aplicativo *mobile* que, posteriormente, será disponibilizado para os principais sistemas de *smartphones*, onde os usuários que se cadastrarem terão acesso a todos os eventos disponíveis, com descrições, datas, horários e valores, podendo, inclusive, realizar a compra de ingressos pela plataforma. A Figura 3 exemplifica graficamente a arquitetura do sistema.

FIGURA 3 – ARQUITETURA DO SISTEMA DESENVOLVIDO



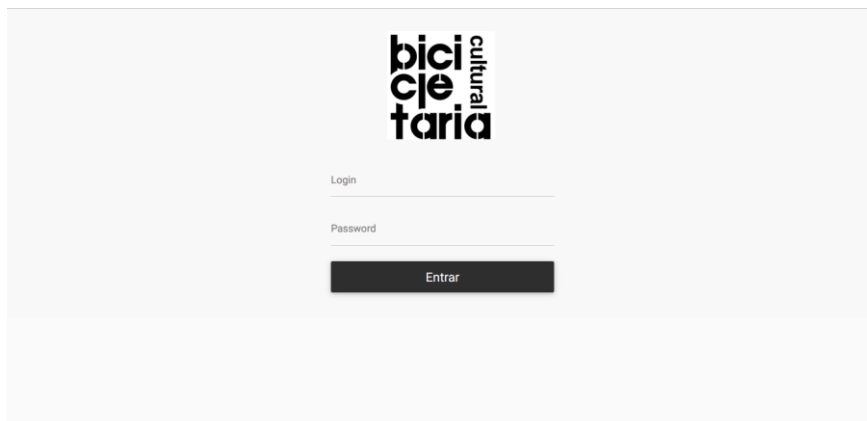
Legenda: Nesta Figura, é apresentado como o sistema é estruturado e como cada tipo de usuário interage com ele de maneira macro. Pode-se ver que sob um servidor *Tomcat* há um software web que serve de BackOffice para os administradores, além de um aplicativo *mobile* utilizado pelos clientes da Bicicletaria Cultural. Ambos se comunicam através de um *Web Service* e acessam um mesmo banco de dados MySQL. Fonte: Os autores.

As seções a seguir apresentam o sistema em detalhes, uma tela de cada vez, com descrição e imagem, explicando o que pode ou deve ser feito a cada etapa.

4.1 APLICAÇÃO WEB

Começando pela aplicação web, é necessário realizar o login para que se possa acessar como mostra a Figura 4. Haverá um usuário “Admin” previamente criado, contudo nosso cliente terá a opção de criar novos usuários que também tenham acesso a plataforma se assim quiser.

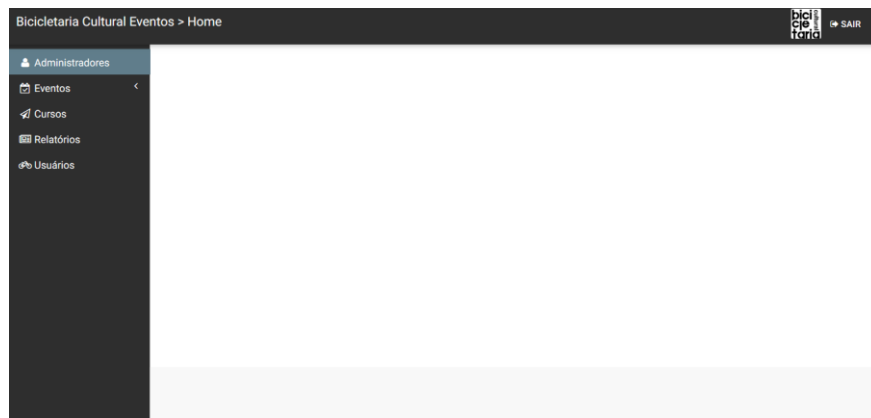
FIGURA 4 – TELA DE LOGIN PARA A APLICAÇÃO WEB DO SISTEMA BICICLETARIA CULTURAL EVENTOS

A imagem mostra a interface de login de uma aplicação web. No topo, há um logotipo com o texto "bici cultural taria" em uma fonte moderna, onde "bici" e "taria" estão em negrito e "cultural" é menor e centralizado entre eles. Abaixo do logotipo, há dois campos de entrada de texto: o primeiro rotulado "Login" e o segundo rotulado "Password". Ambos os campos têm uma borda superior cinza. Abaixo dos campos, há um botão retangular com o texto "Entrar" em branco.

Legenda: Primeira tela apresentada ao administrador quando tenta acessar o sistema. Nada pode ser feito sem que, antes, seja efetuado o login. Fonte: Os autores.

Ao realizar o *login*, o usuário é direcionado a *home* do site, uma página com um menu lateral e um botão para sair e realizar o *logout*, como mostra a Figura 5.

FIGURA 5 – TELA INICIAL DA APLICAÇÃO WEB DO SISTEMA BICICLETARIA CULTURAL EVENTOS



Legenda: Ao *logar* no sistema, é apresentada ao administrador a tela inicial com um menu de navegação no lado esquerdo da tela. Através deste menu, todas as demais telas podem ser acessadas. Fonte: Os autores.

A navegação é realizada pelo menu lateral. Clicando na opção que deseja acessar para ver uma lista do que já foi criado. A primeira opção apresentada é a tela de Administradores. Aqui é possível gerenciar os usuários que terão acesso ao sistema web como mostra a Figura 6.

FIGURA 6 – TELA PARA GERENCIAR ADMINISTRADORES

Gerenciar Administradores

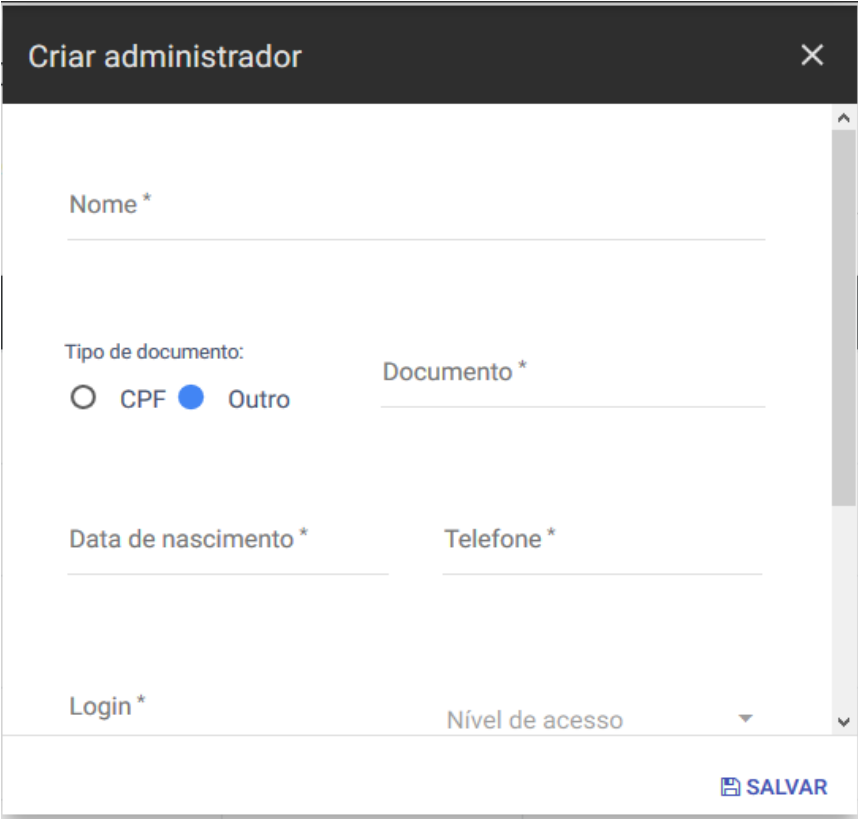
[+ NOVO CADASTRO](#) Pesquisa rápida

Documento	Login	Nível de acesso	Ações
00000000000	sandrini	READ	EDITAR EXCLUIR
1111	leo	FULL	EDITAR EXCLUIR
333	te	FULL	EDITAR EXCLUIR
3332222	the	FULL	EDITAR EXCLUIR
37136333040	admin	FULL	EDITAR EXCLUIR

Legenda: esta tela mostra os administradores já criados, o nível de permissão deles, seus números de documento, um botão para adicionar novos administradores e uma barra de pesquisa para procurar um administrador por palavra-chave. Fonte: Os autores.

Clicando no botão de novo cadastro, temos acesso a um formulário onde inserimos todos os dados necessários para criar um novo registro de administrador. Veja na Figura 7.

FIGURA 7 – FORMULÁRIO DE CRIAÇÃO DE ADMINISTRADORES

A imagem mostra uma interface web para criar um novo administrador. O formulário é encabeçado por uma barra escura com o título "Criar administrador" e um ícone de fechar. O formulário contém os seguintes campos: "Nome *" (obrigatório), "Tipo de documento:" com opções "CPF" (desselecionada) e "Outro" (selecionada), "Documento *" (obrigatório), "Data de nascimento *" (obrigatório), "Telefone *" (obrigatório), "Login *" (obrigatório) e "Nível de acesso" (menu suspenso). Um botão "SALVAR" com um ícone de salvar está no canto inferior direito.

Legenda: nesta Figura, vemos alguns campos necessários para criar um novo administrador. Os campos obrigatórios possuem um asterisco sinalizando. Primeiramente, é preciso informar o nome. Em seguida, deve-se escolher o tipo de documento e informar o número. Após, insere-se a data de nascimento, o telefone, um login, e um nível de acesso. O formulário só pode ser submetido depois que todos os campos estão preenchidos. Fonte: Os autores.

FIGURA 8 – CONTINUAÇÃO DO FORMULÁRIO PARA CRIAÇÃO DE NOVOS ADMINISTRADORES

O formulário 'Criar administrador' possui uma barra superior escura com o título e um ícone de fechar. Abaixo, há uma seção com radio buttons para 'CPF' e 'Outro'. O formulário contém os seguintes campos obrigatórios (indicados por *):

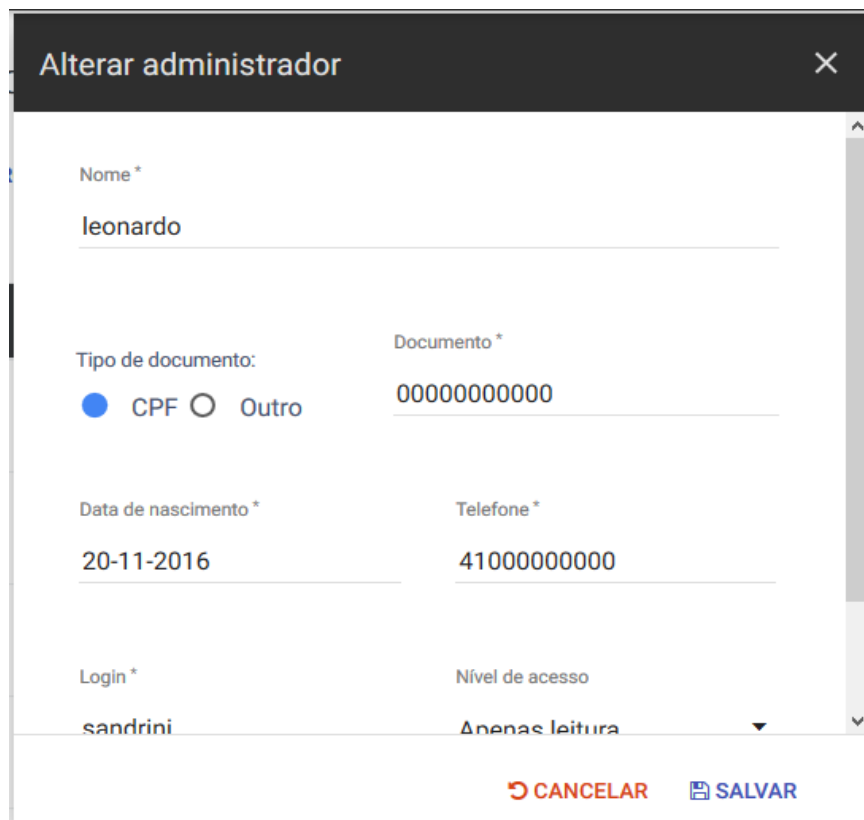
- Data de nascimento ***: Campo de texto.
- Telefone ***: Campo de texto.
- Login ***: Campo de texto.
- Nível de acesso**: Menu suspenso.
- Senha ***: Campo de texto.

Existem também uma opção ☐ **Exibir senha** e um botão **SALVAR** no canto inferior direito.

Legenda: Esta Figura mostra o fim do formulário, deixando claro que também é preciso cadastrar uma senha. Fonte: Os autores.

O formulário de edição apresenta os mesmos campos, porém já preenchidos com os dados informados anteriormente. Para alterar algum dado, basta sobrescrever o campo e clicar em salvar, a exemplo da Figura 9.

FIGURA 9 – FORMULÁRIO PARA EDIÇÃO DE ADMINISTRADOR



Alterar administrador

Nome *

leonardo

Tipo de documento:

☒ CPF ☐ Outro

Documento *

00000000000

Data de nascimento *

20-11-2016

Telefone *

41000000000

Login *

sandrini

Nível de acesso

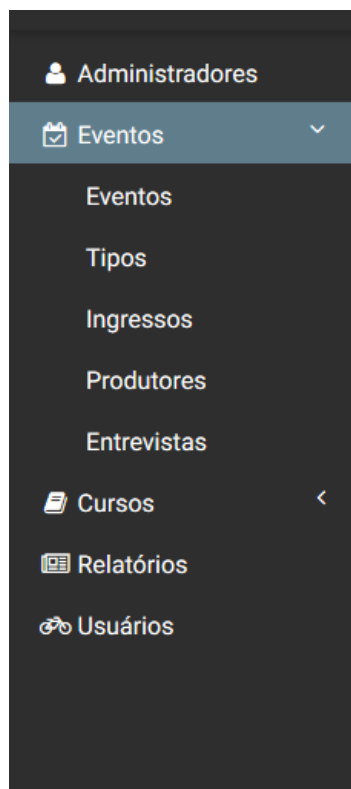
Apenas leitura

CANCELAR SALVAR

Legenda: a Figura acima apresenta um formulário idêntico ao primeiro, porém já preenchido.
Fonte: os autores.

A opção escolhida no menu apresenta cor diferente dos demais campos para salientar a seleção, a exemplo da Figura 10.

FIGURA 10 – MENU SELECIONADO



Legenda: Esta Figura mostra a opção do menu “Eventos” em destaque. Ao selecionar essa opção, é exibido um submenu com novas opções. Fonte: Os autores.

Todas as telas apresentam o mesmo padrão de estrutura, com itens criados já listados, um botão para adicionar novos itens e botões ao lado de cada item para editar e excluir. O que os difere são os dados a serem inseridos nos formulários.

O Formulário para criação de novo evento tem uma série de informações para serem adicionadas obrigatoriamente, como pode ser visto nas próximas Figuras.

FIGURA 11 – INFORMAÇÕES GERAIS DO FORMULÁRIO DE EVENTOS

The screenshot shows a web form titled 'Novo Evento' with a close button (X) in the top right corner. The form contains the following fields:

- Nome *** and **Subtítulo ***: Two text input fields at the top.
- Tipo de evento ***: A dropdown menu.
- Produtor ***: A dropdown menu.
- Data *** and **Horário ***: Two text input fields.
- Descrição ***: A large text area for a detailed description.
- Artistas convidados *** and **FAQ**: Two text input fields at the bottom.

At the bottom right of the form, there are two buttons: **CANCELAR** (with a red icon) and **SALVAR** (with a save icon).

Legenda: Nestes campos, o administrador deve informar alguns dos dados principais para a criação de um novo evento, como o nome, um subtítulo, a data e hora e uma descrição geral do que o evento se tratará. Fonte: Os autores.

FIGURA 12 – INFORMAÇÕES DE ENDEREÇO DO FORMULÁRIO DE EVENTOS

The screenshot shows the same 'Novo Evento' form, but with the address and ticket sections expanded. The fields include:

- Endereço**: A dropdown menu.
- CEP ***, **Nome**, and **Rua ***: Three text input fields.
- Bairro ***, **Número ***, **Complemento**, **UF *** (dropdown), and **Cidade *** (dropdown): Five text input fields.
- Ingressos**: A dropdown menu.

The **CANCELAR** and **SALVAR** buttons are still present at the bottom right.

Legenda: Aqui, o administrador deve aprofundar as informações a respeito do local do evento, informando coisas como o CEP, rua, bairro, número e complemento. Fonte: Os autores.

FIGURA 13 – INFORMAÇÕES SOBRE INGRESSOS DO FORMULÁRIO DE EVENTOS

Legenda: Nessa etapa do formulário, o usuário pode escolher uma data e hora para início e fim da venda dos ingressos, além de definir um tipo, uma classificação cultural do evento, a quantidade de ingressos e o valor cobrado. Fonte: Os autores.

FIGURA 14– DETALHES DO FORMULÁRIO DE EVENTOS

Legenda: Nesta parte do formulário o administrador pode definir uma série de detalhes mais aprofundados sobre o evento como a taxa de ocupação, de bilhete, a data de início e fim de uma produção. Também é possível escolher coisas como o tamanho da equipe formada ou o formato do evento. Fonte: Os autores.

FIGURA 15 – CONTINUAÇÃO DA SESSÃO DE DETALHES DO FORMULÁRIO DE EVENTOS

The screenshot shows a web form titled 'Novo Evento' with a close button (X). The form is divided into six sections, each with a label and a corresponding text input field:

- Promoções
- Consumação
- Texto de apresentação
- Material de propaganda
- Equipamento
- Notas pós evento

At the bottom right of the form, there are two buttons: 'CANCELAR' (with a red arrow icon) and 'SALVAR' (with a save icon).

Legenda: Há ainda outros fatores que podem ser registrados na criação de um novo evento, como promoções, consumação e equipamento, por exemplo. Esses campos são todos alternativos para o preenchimento. Fonte: Os autores.

FIGURA 16 – IMAGENS PARA CAPA E DIVULGAÇÃO DOS EVENTOS

The screenshot shows the 'Novo Evento' form with the 'Imagens' section expanded. It includes three dropdown menus: 'Ingressos', 'Detalhes da Produção', and 'Imagens'. Below the 'Imagens' dropdown, there are two rows of input fields:

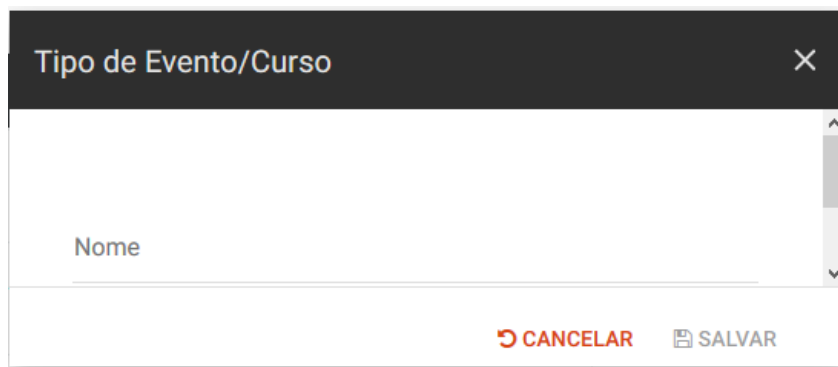
- Banner:** A button 'Escolher arquivo' followed by a text field labeled 'Banner do evento *'.
- Divulgação:** A button 'Escolher arquivo' followed by a text field labeled 'Imagem de divulgação'.

At the bottom right, the 'CANCELAR' and 'SALVAR' buttons are visible.

Legenda: Finalmente, neste formulário é possível escolher imagens para ficarem no topo do evento e servirem como divulgação. Fonte: Os autores.

Cada uma das seções possuem os seus próprios formulários para que o administrador tenha liberdade do que decidir criar de novo no sistema. Abaixo dos eventos, são listados os tipos de eventos e cursos e este também possui seu próprio formulário, que pode ser visto na Figura 17.

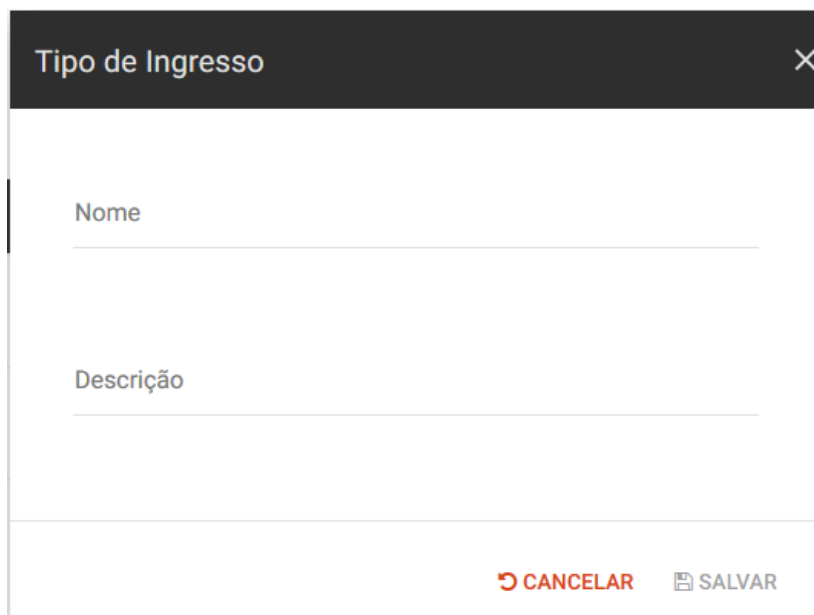
FIGURA 17 – FORMULÁRIO DE TIPO DE EVENTO/CURSO

A imagem mostra uma interface de usuário para um formulário. No topo, há uma barra escura com o título "Tipo de Evento/Curso" em branco e um ícone de fechar (X) no canto superior direito. Abaixo da barra, há um campo de texto grande e vazio com o rótulo "Nome" em cinza. No canto superior direito do campo de texto, há uma seta para cima (^) e uma seta para baixo (v). Na base do formulário, há uma barra de ação com dois botões: "CANCELAR" em vermelho com um ícone de seta curva e "SALVAR" em cinza com um ícone de disquete.

Legenda: Este formulário possui um único campo com o nome do tipo de evento ou curso. Fonte: Os autores.

Além dos tipos diferentes de eventos e curso, o administrador também pode criar tipos diferentes de ingresso através do formulário apresentado na Figura 18.

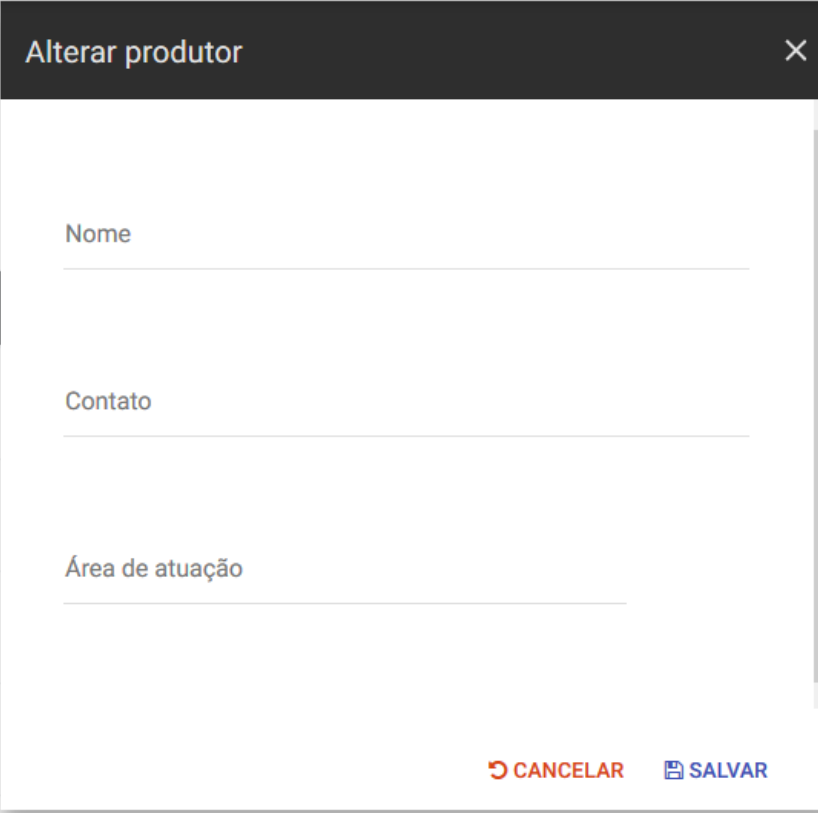
FIGURA 18 – FORMULÁRIO DE TIPO DE INGRESSO

A imagem mostra uma interface de usuário para um formulário de "Tipo de Ingresso". O formulário é apresentado em uma caixa modal com um cabeçalho escuro contendo o título "Tipo de Ingresso" e um ícone de fechar (X). O corpo do formulário possui dois campos de texto: "Nome" e "Descrição", cada um com uma linha de entrada. Na base da caixa, há dois botões: "CANCELAR" com um ícone de seta circular vermelha e "SALVAR" com um ícone de disquete azul.

Legenda: Assim como o formulário anterior, este também é bem simples, pedindo apenas um nome e uma descrição. Fonte: Os autores.

Outro cadastro possível é o de produtores, as pessoas que realizam os eventos e cursos promovidos pela aplicação *mobile*. A Figura 19 mostra o como um novo item pode ser cadastrado nesta lista.

FIGURA 19 – FORMULÁRIO PARA CRIAÇÃO DE NOVO PRODUTOR

A imagem mostra uma interface web para alterar um produtor. No topo, há uma barra escura com o título "Alterar produtor" em branco e um ícone de fechar (X) à direita. Abaixo, há três campos de texto com rótulos "Nome", "Contato" e "Área de atuação" em uma fonte cinza. Cada rótulo está alinhado à esquerda do seu respectivo campo de entrada. No rodapé da interface, há dois botões: "CANCELAR" com um ícone de seta curva vermelha e "SALVAR" com um ícone de disquete azul.

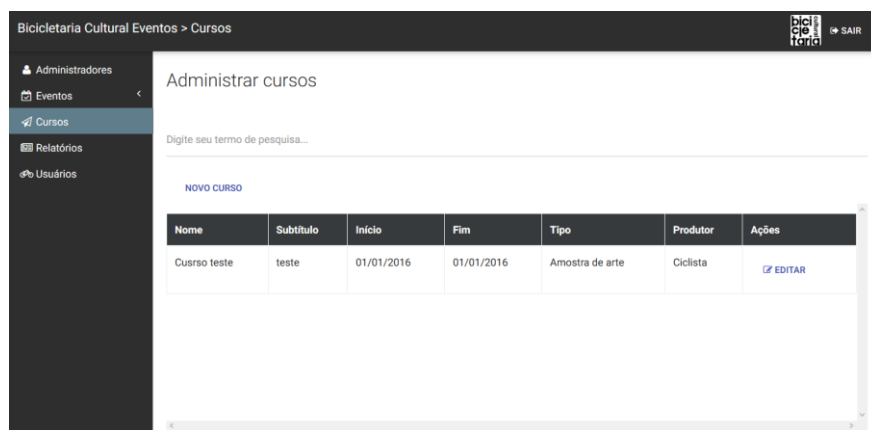
Legenda: De forma semelhante aos últimos, este também é um formulário bem simples, apresentando apenas um campo de nome, um de contato e um campo onde deve ser informada a área de atuação do produtor.

Clicando na opção “Cursos” como apresentado na Figura 20, o usuário tem acesso a todos os cursos já criados em uma lista. Cada item desta lista representa um curso e possui um título, um subtítulo, uma data de início e fim do curso, um tipo para o curso, o nome do produtor responsável e um botão editar, que dá a possibilidade de alterar as informações do curso criado.

Também é possível ver nessa tela, assim como em outras deste sistema, uma barra de busca superior, que permite procurar um item específico da lista através de palavras chave.

Abaixo da barra de busca, há um botão “NOVO CURSO”, que dá a opção de criar um novo item.

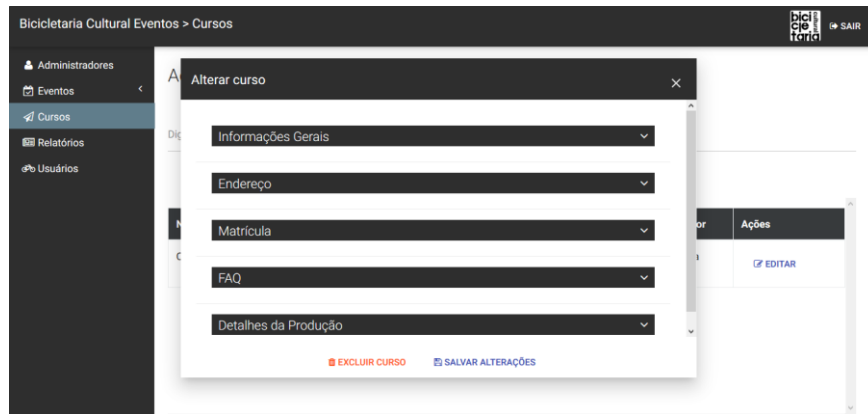
FIGURA 20 – TELA PARA ADMINISTRAÇÃO DE CURSOS



Legenda: aqui é possível ver a lista de cursos já criados, apresentados por nome, subtítulo, data de início e fim, tipo de curso e quem o produziu. As ações possíveis para o usuário são de clicar no botão “EDITAR” no canto direito, em “NOVO CURSO”, logo acima da lista, para adicionar um novo curso ou, realizar uma busca entre os cursos já existentes na barra de busca no topo da tela, abaixo da descrição da tela. Fonte: Os autores.

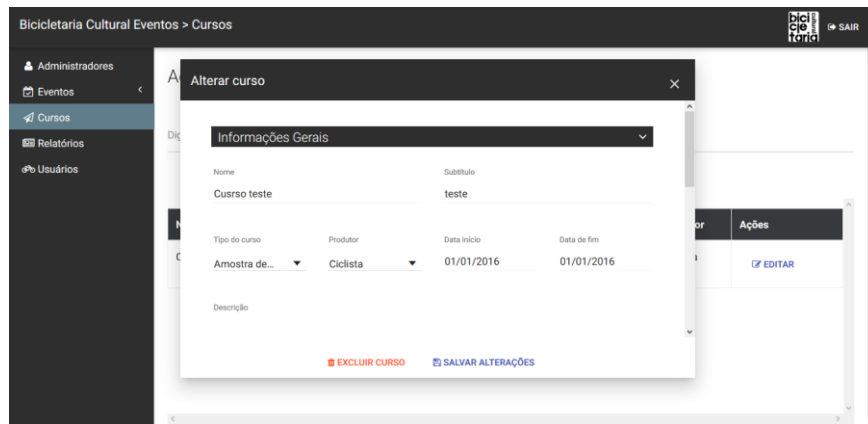
Nas Figuras 21 e 22, pode-se ver como o sistema se comporta ao se tentar alterar um item. Uma caixa de diálogo que apresenta os mesmos campos mostrados para criar um item novo, porém com as opções “EXCLUIR CURSO” e “SALVAR ALTERAÇÕES”.

FIGURA 21 – CAIXA DE DIÁLOGO APRESENTANDO OS CAMPOS PARA EDIÇÃO RETRAÍDOS EM CONJUNTOS



Legenda: Nesta tela, é aberta uma grande caixa de diálogo com um formulário onde o administrador informa, em "Informações Gerais", uma descrição mais genérica do curso. Há um campo "Endereço", onde é informado o local do curso. "FAQ" é o campo onde o administrador preenche com perguntas realizadas frequentemente e suas respostas já definidas. E a um campo "Detalhes da Produção", em que quaisquer considerações mais aprofundadas sobre o curso podem ser feitas. Fonte: Os autores.

FIGURA 22 – CAIXA DE DIÁLOGO PARA EDIÇÃO DE CURSO APRESENTANDO O CONJUNTO "INFORMAÇÕES GERAIS" EXPANDIDO



Legenda: Ao expandir o campo "Informações Gerais", o administrador deve definir um nome para o curso, um subtítulo, um tipo de curso, quem é o produtor, a data de início e fim e a descrição geral do curso. Fonte: Os autores.

Já clicando no botão "NOVO CURSO" o mesmo formulário é aberto, porém tomando a tela toda como mostra a Figura 23.

FIGURA 23 – TELA PARA CADASTRO DE NOVO CURSO

Legenda: esta tela apresenta um formulário idêntico ao de edição de cursos. Contudo, este, ao ser salvo, cria um novo item do tipo curso e é acessado quando o administrador clica no botão "NOVO CURSO". Fonte: Os autores.

Outras opções como "Administradores" e "Usuários" oferecem as mesmas opções para quem estiver logado. Já a aba eventos mostra algumas opções a mais, como fica visível na Figura 24. Ao clicar em "Eventos" são apresentados novos botões para acessar listas de "Eventos", "Produtores" e "Entrevistas", além de dar uma opção para "Cadastrar eventos".

FIGURA 24 – OPÇÃO EVENTOS EXPANDIDA, MOSTRANDO A SUBOPÇÃO "CADASTRAR EVENTOS" SELECIONADA, QUE ABRE UM FORMULÁRIO PARA CADASTRO DE NOVO EVENTO

Legenda: muito semelhante aos demais formulários, este é utilizado para o cadastro de novos

itens do tipo “Evento” é acessado diretamente pelo menu de navegação lateral quando o administrador clica em “Eventos” e, em seguida, em “Cadastrar eventos”.

Em seguida, serão apresentadas as telas da outra face deste sistema, desenvolvida para dispositivos móveis.

4.2 APLICAÇÃO MOBILE

A aplicação *mobile* tem uma interface de *login* muito semelhante à versão web, como apresentado na Figura 25, porém esta parte do sistema tem foco no uso dos clientes da empresa Bicletaria Cultural.

FIGURA 25 – TELA DE LOGIN



Legenda: Tela inicial para acesso a aplicação *mobile*. Fonte: Os autores.

Caso seja um novo usuário, pode-se realizar o cadastro a partir do link apresentado na tela de *login* escrito “Cadastre-se”. Uma tela com formulário, como o da Figura 26, é apresentada ao usuário para que se registre no aplicativo.

FIGURA 26 – FORMULÁRIO PARA CADASTRO DE USUÁRIOS DO APLICATIVO

The image shows a mobile application interface for user registration. At the top, there is a status bar with icons for signal, battery, and time (14:27). Below the status bar is a header bar with a back arrow and the title "Cadastro". The form consists of several input fields: "Nome completo", "Data de nascimento", "Telefone", "Tipo de documento" (with two radio buttons labeled "CPF" and "Outro"), "Documento", "Email", "CEP", "Estado" (with a dropdown arrow), and "Selecione" (with a dropdown arrow). At the bottom, there is a navigation bar with three icons: a back arrow, a home icon, and a square icon.

Legenda: O formulário mostra um campo para informar o nome do usuário, sua data de nascimento e um telefone. Também pede o número de um documento, dando a opção de um CPF ou algum outro, um email, o endereço e uma senha. Fonte: Os autores.

Ao acessar o sistema, são visíveis os documentos criados pelos administradores em um feed. É possível ver alguns dados do evento como uma descrição, uma data e um horário, como é apresentado pela Figura 27.

FIGURA 27 – TELA DE EVENTOS



Legenda: a tela apresenta detalhes do evento como um nome, data, horário, uma descrição e uma imagem ilustrativa do que se trata o evento. Fonte: Os autores.

Também, ao acessar um evento em detalhe, é possível realizar a compra de ingressos para participar dentro do próprio aplicativo. Ao fim da tela de detalhes do aplicativo, assim como na Figura 28, apresenta-se o endereço do evento, seu valor e um botão “Comprar”.

FIGURA 28 – TELA DE DETALHES DO EVENTO – BOTÃO COMPRAR



Legenda: Tela de detalhes do evento, onde também é possível realizar a compra de ingressos para o mesmo. Fonte: Os autores.

Ao optar por clicar no botão de compra, o usuário é direcionado para uma tela de pagamento, onde pode escolher entre duas formas: pagar com *Paypal* ou com cartão de crédito, igual ao que é mostrado pela Figura 29.

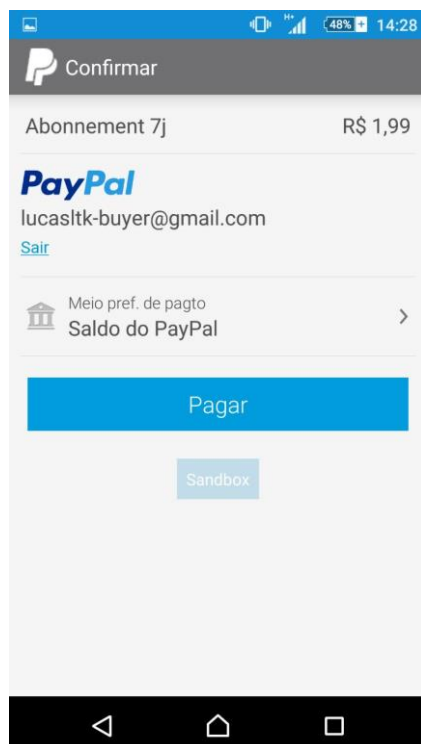
FIGURA 29 – TELA DE OPÇÕES DE PAGAMENTO



Legenda: Tela para selecionar a forma de pagamento do ingresso. Fonte: Os autores.

Por exemplo, ao selecionar a opção de pagar com o *Paypal*, é exibida uma nova tela, em que você pode realizar o pagamento do ingresso com o saldo já disponível em sua conta, tal qual é mostrado na Figura 30.

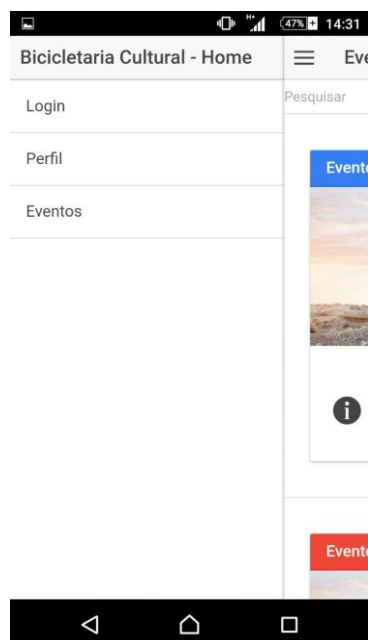
FIGURA 30 – TELA PARA PAGAMENTO COM PAYPAL



Legenda: a tela mostra um meio de encerrar uma comprar através do aplicativo realizado o pagamento com o *Paypal*. Fonte: Os autores.

A partir de um menu lateral, visível na Figura 31, é possível ter acesso a links rápidos, como o perfil do usuário e a tela de eventos.

FIGURA 31 – MENU LATERAL



Legenda: Menu lateral para acessos rápidos. Fonte: Os autores.

O Capítulo seguinte traz uma conclusão pra esse documento, onde é avaliado o resultado final, buscando entender se o objetivo traçado de início foi alcançado e quais foram os pontos positivos e negativos encontrados ao longo deste projeto.

5 CONCLUSÃO

5.1 CONSIDERAÇÕES QUANTO AO PROJETO

Ao início deste projeto, foram planejadas diversas funcionalidades para atender ao cliente de forma plena. Contudo, as dificuldades em manter os prazos de desenvolvimento paralelamente a uma linha de aprendizado de diversas tecnologias, até então, pouco conhecidas pelos acadêmicos envolvidos, não permitiu que o sistema planejado de princípio, fosse concretizado até a data final de entrega.

Contudo, buscara-se apresentar a ferramenta mais completa possível, com todas as funcionalidades essenciais implementadas e livres de *bugs*. Focou-se no que seria utilizado de imediato pelo cliente, visando implementar tudo o que não fora finalizado em uma nova fase deste projeto, trabalhando com prazos, agora, estabelecidos pelos clientes. O plano é que a ferramenta receba atualizações com novas funcionalidades até atingir sua versão final, onde atenda todos os requisitos do cliente estabelecidos de início, ou mesmo, ao longo do projeto.

A princípio, quando alcançada a data de entrega final do presente projeto, os clientes terão como administrar e promover seus eventos com um sistema web, e seus clientes poderão baixa-lo através da *Play Store*, a loja de aplicativos do *Android*. A versão para dispositivos *Apple* também será disponibilizada posteriormente. Por termos trabalhado com uma tecnologia *cross-mobile*, o *Ionic*, não há um processo de migração por completo, sendo necessária somente uma pequena adaptação feita em um sistema *MacOS*, para que a aplicação mobile seja executável em dispositivos com sistema *IOS*. Sendo assim, o espaço entre o lançamento da versão *Android* para a *IOS* não trará impacto à utilização do *software*.

5.2 PONDERAÇÕES QUANTO AOS ASPECTOS POSITIVOS E NEGATIVOS

Este projeto trouxe diversas conquistas aos acadêmicos envolvidos em diversos aspectos.

Primeiramente, apresentando a todos um clima real de projeto de desenvolvimento de *software*, uma experiência que aos envolvidos não apenas na busca pela conclusão do curso tecnológico de graduação em Análise e Desenvolvimento de Sistemas, como também no ingresso ao mercado de trabalho, permitindo apresentar uma carga de experiência vivida na prática em âmbito acadêmico.

Também, muitos conhecimentos tecnológicos foram assimilados por todos, tendo que buscar recursos nunca antes usados e aprendê-los a tempo de concluir cada nova etapa do processo de desenvolvimento. A própria metodologia ágil para gerenciamento das atividades, o *Scrum*, ainda que não tenha sido aplicada de forma profunda, trouxe uma visão prática do que há no mercado atual em relação a gestão de projetos.

Analisando todos os ganhos dos acadêmicos que atuaram neste projeto, considera-se improdutivo apenas o tempo de desenvolvimento, ocasionado justamente pela pouca vivência dos envolvidos com projetos e com as tecnologias aplicadas a este. Está, porém, foi uma adversidade superada de forma gradativa, numa linha de aprendizado que ascendeu em paralelo às atividades desempenhadas no projeto.

5.3 CONSIDERAÇÕES FINAIS

Avaliamos como alcançado o objetivo principal deste projeto, entregando um sistema que atende os requisitos prioritários de nosso cliente. Num espectro mais específico, nem todas as metas foram cumpridas, contudo, por avaliar-se que, por menores que sejam as funcionalidades, todas são essenciais ao cliente do projeto, foi decidido dar continuidade ao desenvolvimento em uma nova etapa, até que produto final atenda a todos os requisitos primariamente estabelecidos.

6 REFERÊNCIAS

LAUDON, K.; LAUDON, J. **Management Information Systems 7th Edition**, 2014.

Sippu, S.; Soisalon-Soininen, E.; **Transaction Processing: Management of The Logical Database and its Underlying Physical Structure**, 2015.

RAVULAVARU, A.; HARTINGTON, M. **Learning Ionic: Build real-time and hybrid applications with Ionic**, 2015.

ALMEIDA, F. **MEAN – Full stack JavaScript para aplicações web com MongoDB, Express, Angular e Node**, 2015.

BECK, K.; BEEDLE, M.; BENNEKUM, A. V.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS D. **Manifesto Ágil**, 2001.

COHN, M. **Succeeding with Agile – Software Development Using Scrum**, 2010.

SABBAGH, R. **Scrum – Gestão Ágil para Projetos de Sucesso**, 2013.

FOWLER, M. **UML Essencial – 3ª Edição**, 2005.

6.1 APÊNDICE A – REQUISITOS FUNCIONAIS

Abaixo, listaremos todos os requisitos funcionais do sistema, informando o código do requisito, seu nome, entre parênteses o tipo de usuário ao qual se aplica, e, por fim, uma descrição do que o cliente espera que seja possível a partir daquele requisito.

Os tipos de usuário serão representados pelas siglas CLI para os clientes da Bicicletaria Cultural e ADM para os administradores. Caso seja um requisito não aplicável a um tipo de usuário em específico, será seguido por “N/A”.

RF01 – Cadastro de administrador (ADM)

Um administrador pode acessar a plataforma *web* e realizar o cadastro de um novo administrador.

RF02 – Cadastro de usuário (CLI)

Um usuário pode acessar o aplicativo de eventos e realizar seu cadastro para acessar todas as funcionalidades.

RF03 – Cadastro de eventos (ADM)

Um administrador pode, através da plataforma *web*, criar um evento novo para ser exibido no aplicativo.

RF04 – Cadastro de cursos (ADM)

O sistema deve fazer uma diferenciação entre cursos e eventos, dando a possibilidade de cadastrar ambos em formulários específicos para cada um.

RF05 – Cadastro de produtores (ADM)

Os administradores podem cadastrar produtores diferentes deles próprios para os cursos e eventos cadastrados no *app*.

RF06 – Cadastro de entrevistas (ADM)

Os administradores podem cadastrar conteúdos de entrevistas para serem exibidas no aplicativo.

RF07 – Cadastro de diferentes sessões (ADM)

Os eventos e cursos podem se repetir em diferentes sessões. A quantidade de sessões e horários devem ser informadas.

RF08 – Edição e exclusão de cadastros (ADM)

Tudo o que for cadastrado deve permitir edição dos dados informados e a exclusão por completo de um cadastro.

RF09 – Visualização de eventos e cursos (CLI)

A visualização de eventos e cursos deve ser possível por todos que acessarem o aplicativo, mesmo que não tenham realizado *login*.

RF10 – Visualização de entrevistas (CLI)

O aplicativo deve apresentar as entrevistas cadastradas pelos administradores na plataforma *web*.

RF11 – Compra de ingressos (CLI)

Deve ser possível, para todos os usuários que estejam *logados*, a compra de ingressos para os eventos que visualizarem.

RF12 – Comentários nos eventos e cursos (CLI)

Para todos os clientes que tenham realizado *login* e a compra de ingressos deve haver a possibilidade comentar nos eventos e cursos exibidos no aplicativo.

6.2 APÊNDICE B - REQUISITOS NÃO-FUNCIONAIS

RNF01 – Identidade do sistema (N/A)

O sistema deve usar as mesmas cores já adotadas pela empresa Bicicletaria Cultural e também suas imagens para *logo*. O aplicativo deve ter o mesmo nome da empresa seguido pela palavra “Eventos”.

RNF02 – Segurança (N/A)

As senhas para acesso ao sistema são todas criptografadas. O acesso ao sistema administrativo só é possível através de login e apenas um administrador

cadastrado pode realizar o cadastro de outro. O acesso ao banco também é restrito a administradores.

RNF03 – Feedback (N/A)

Para os clientes *logados*, após a realização dos eventos aos quais tenham participado, deve ser possível a avaliação através de algum recurso gráfico simples de nota.

Também, os clientes que participaram dos eventos podem enviar, através do aplicativo, suas impressões finais para que os administradores avaliem.

RNF04 – Avaliação de comentários (N/A)

Os comentários feitos no evento devem ser exibidos de imediato, porém podem ser excluídos por um dos administradores caso seja considerado ofensivo ou desrespeite qualquer política estabelecida pelos administradores.

RNF05 – Ubiquidade (N/A)

A um padrão na organização dos componentes entre as interfaces para fácil aprendizado de uso dos usuários sem que haja falhas operacionais.

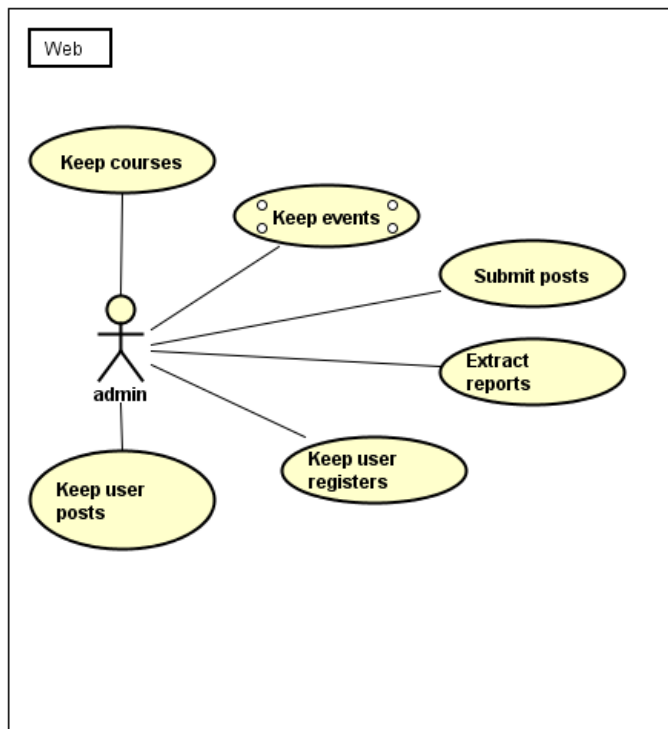
6.3 APÊNCICE C - DIAGRAMAS DE CASOS DE USO

Os diagramas de casos de uso elaborados dividem as interações entre os softwares *mobile* e *web*, cada um tendo como foco um ator diferente. No sistema web, o ator é o administrador e no mobile, o cliente.

O primeiro diagrama a ser apresentado é o do administrador utilizando o sistema web que serve como um *BackOffice* para gerenciar a aplicação *mobile*.

Como mostra a Figura 31, no diagrama podemos ver as relações do ator, neste caso o administrador, através do sistema web, mantendo os cadastros de dos usuários participantes dos eventos e cursos, de eventos, de cursos, de outros administradores e o seu próprio. Além disso, mantém as postagens dos usuários e pode submeter as suas próprias. Também pode extrair relatórios.

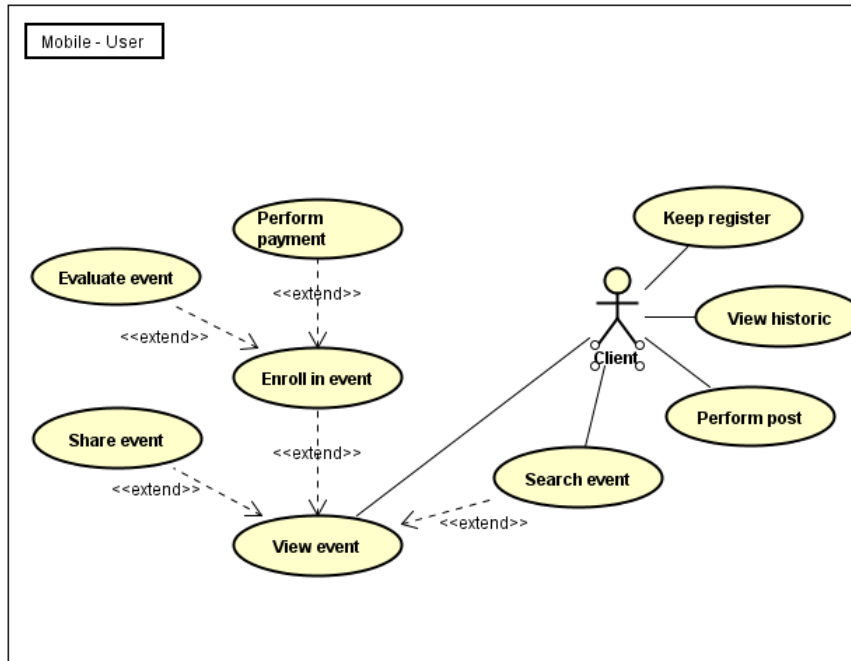
FIGURA 31 – DIAGRAMA DE CASOS DE USO DA APLICAÇÃO WEB



Legenda: a imagem acima apresenta as interações do usuário administrador no sistema web.
 Fonte: Os autores.

O segundo diagrama de casos de uso mostra as interações de um usuário cliente com a aplicação para dispositivos móveis. Através do aplicativo, o cliente pode ver a lista de eventos disponíveis e acessar um evento em detalhe para saber todas as suas informações. Estando *logado*, o usuário pode se inscrever no evento e realizar o pagamento através do próprio *app*. Se quiser, também pode compartilhar o evento em suas redes sociais e, depois da realização, pode avaliá-lo. Também, o cliente pode manter o seu registro, visualizar seu histórico de eventos, realizar postagens e buscar por um evento específico através de palavras-chave.

FIGURA 32 – Diagrama de Casos de Uso da aplicação *mobile*



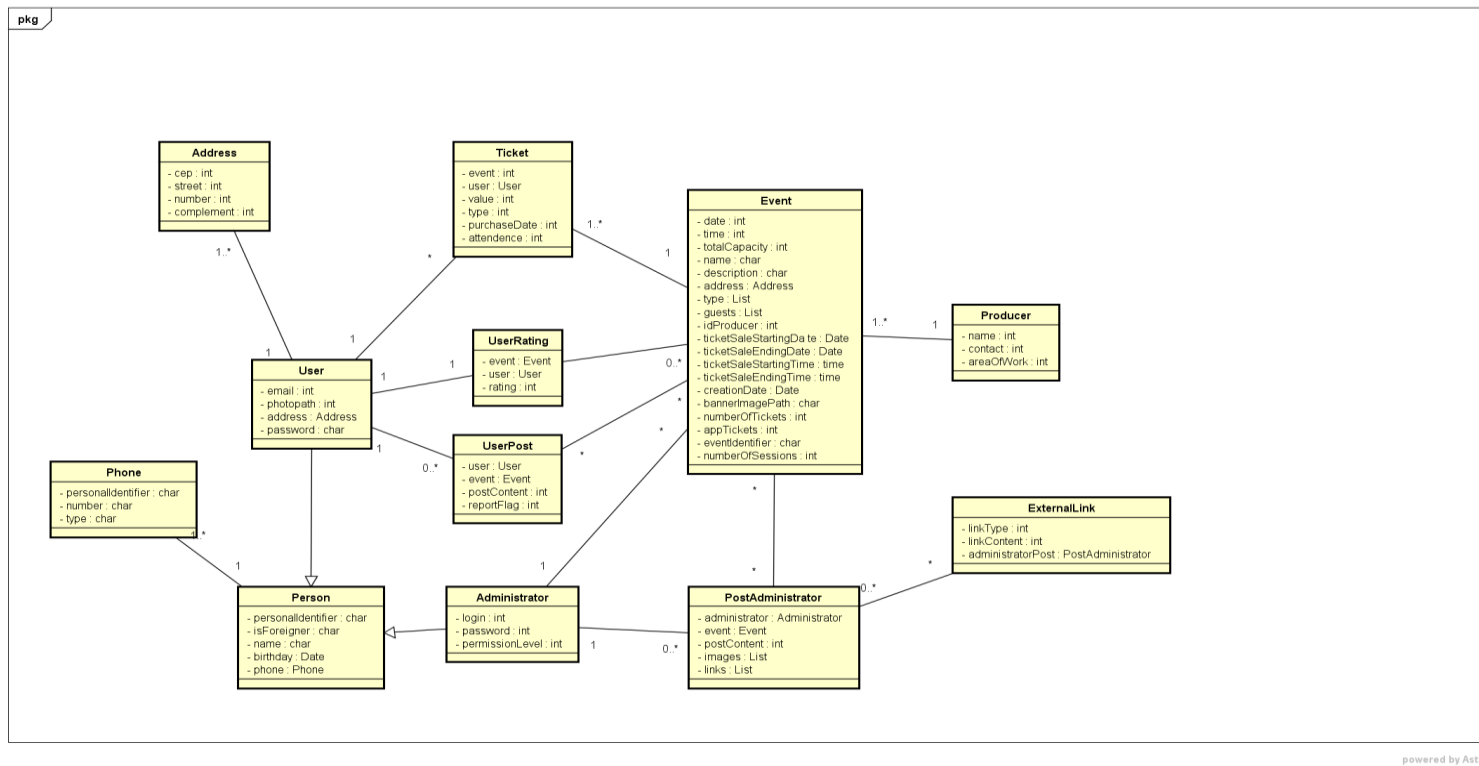
Legenda: o diagrama acima mostra as interações possíveis por um cliente dentro da aplicação para dispositivos móveis. Fonte: Os autores.

6.4 APÊNCICE D - DIAGRAMA DE CLASSES DE ANÁLISE

O diagrama acima mostra a elaboração das classes realizada ao início do projeto. Este diagrama apoiou o desenvolvimento para que a equipe tivesse uma visão completa da estrutura da aplicação, quais as classes necessárias e como deveriam estar organizadas. A classe evento serviu como ponto de partida, por ser o foco de todas as demais funcionalidades presentes na aplicação.

O diagrama desenhado aqui aplica-se às duas ferramentas presentes no projeto, pois apesar das diferentes plataformas, ambas se comunicam como um único sistema.

FIGURA 33 – DIAGRAMA DE CLASSES DE ANÁLISE



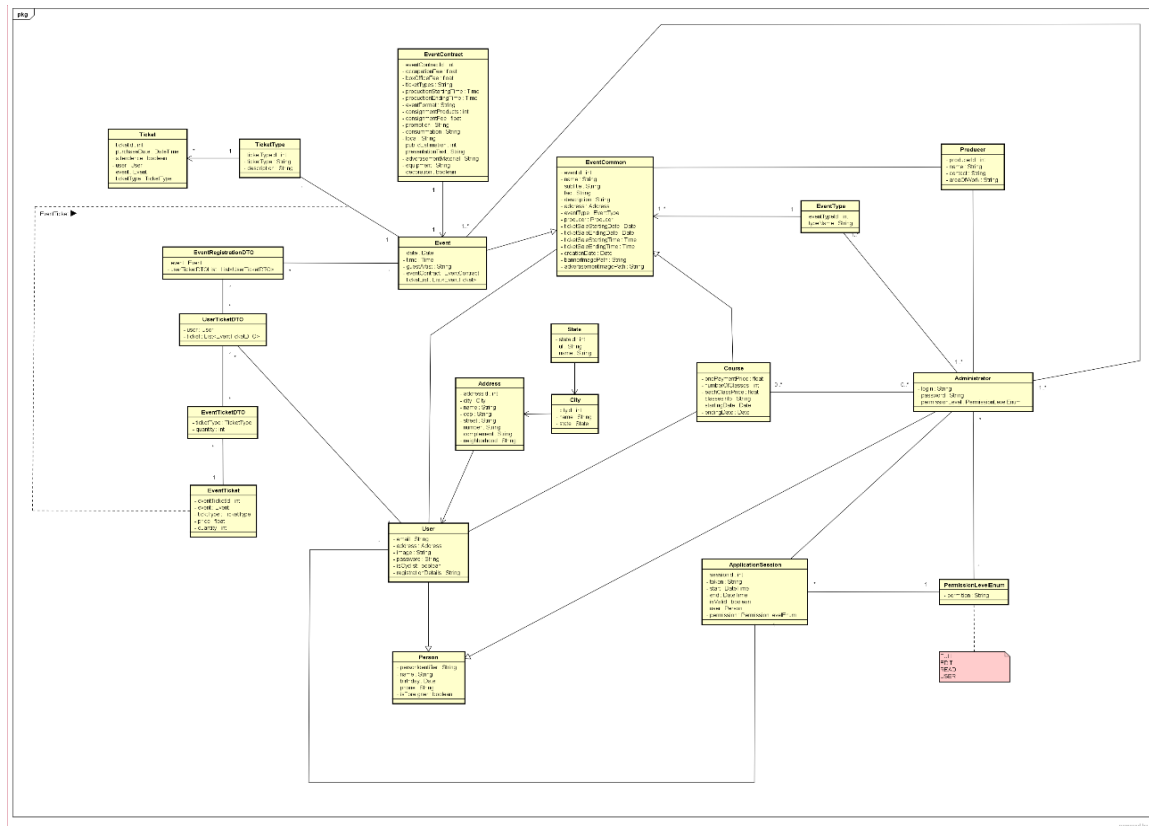
powered by Astah

Legenda: Diagrama de classes elaborado ao início do projeto para apoio do desenvolvimento lógico. Fonte: Os autores.

6.5 APÊNDICE E – DIAGRAMA DE CLASSES DE IMPLEMENTAÇÃO

Este diagrama apresenta a elaboração final das classes, contando com todas as entidades que, de fato, foram aplicadas ao sistema. Diversas classes passaram por mudanças, algumas foram eliminadas do plano final, enquanto outras foram adicionadas ao resultado real.

FIGURA 34 – DIAGRAMA DE CLASSES DE IMPLEMENTAÇÃO



Comentado [RF2]: Precisa melhorar a resolução desta figura. Gere-a numa boa resolução em um arquivo e use o recurso do Word de Inserir -> Figura. Normalmente Ctrl C e Ctrl V não fica legal.

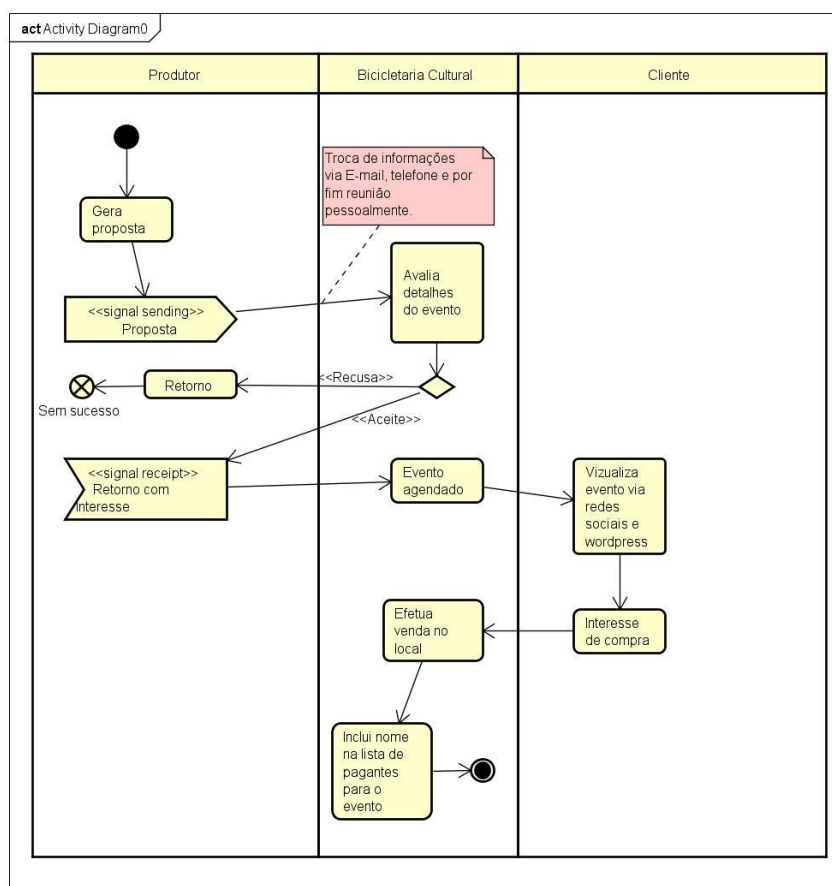
Legenda: o diagrama mostra a estrutura do sistema como realmente foi implementado. Mudanças foram necessárias para a finalização do projeto. Fonte: Os autores.

6.6 APÊNDICE F – DIAGRAMA DE ATIVIDADES

O diagrama abaixo mostra como funcionava o processo de eventos dentro da empresa Bicicletaria Cultural antes do sistema ser implementado.

As etapas deste processo foram informadas pelos clientes do projeto e deram uma visão de como o sistema deveria ser desenvolvido para otimizar as atividades desempenhadas por eles.

FIGURA 35 – DIAGRAMA DE ATIVIDADES

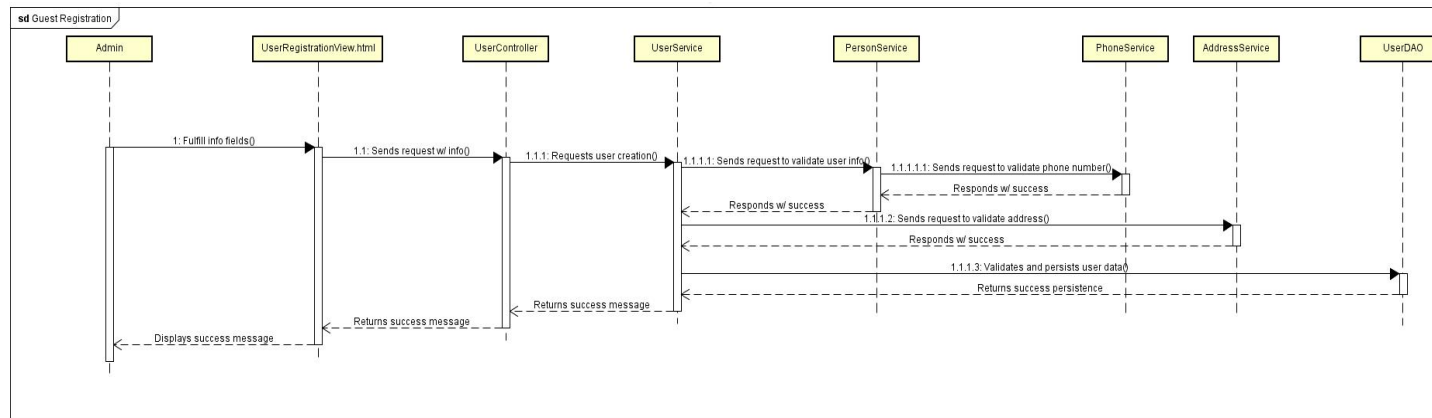


Legenda: Diagrama de atividades com o processo anterior ao sistema. Fonte: Os autores.

6.7 APÊNDICE G – DIAGRAMA DE SEQUÊNCIA DE REGISTRO DE USUÁRIO

Este diagrama tem o propósito de ilustrar o processo de cadastro de um usuário no aplicativo, partindo das entradas em um formulário apresentado na interface gráfica, passando por todas as camadas lógicas em *background* e voltando com um resultado exibido na interface gráfica, como mostra a Figura 36.

FIGURA 36 – DIAGRAMA DE SEQUÊNCIA DO REGISTRO DE USUÁRIO



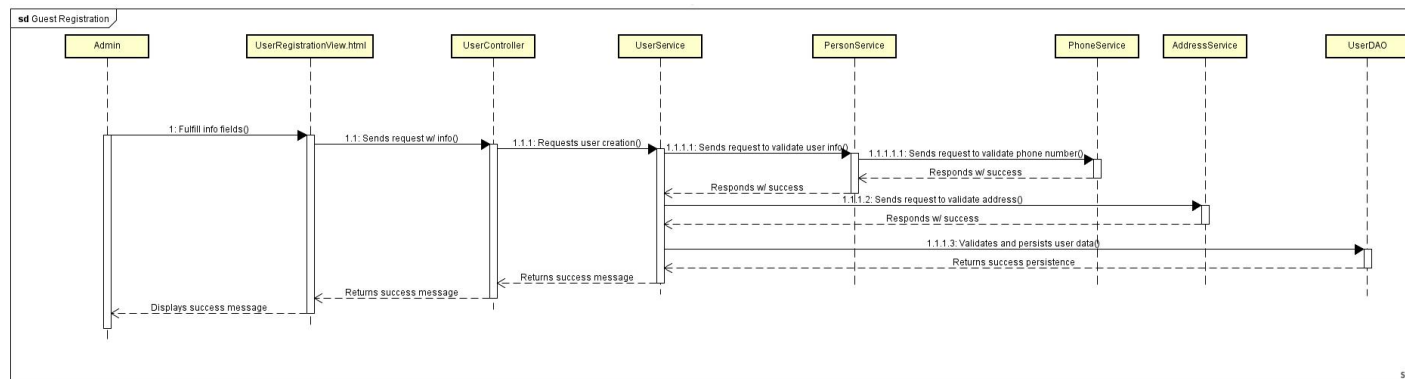
Legenda: O diagrama mostra as etapas de processo e lógica percorridas para um cadastro de usuário. Fonte: Os autores.

6.8 APÊNDICE H – DIAGRAMA DE SEQUÊNCIA DO REGISTRO DE EVENTOS

Muito semelhante ao diagrama de sequência anterior, este segue passos muito parecidos na forma como as etapas se desenrolam. Contudo, por tratar informações diferentes, este diagrama também apresenta outros *webservices* processando os dados informados pelo usuário que, neste caso é o administrador do sistema.

Comentado [RF3]: Cada apêndice em uma nova página

FIGURA 37 – DIAGRAMA DE SEQUÊNCIA DO CADASTRO DE EVENTOS



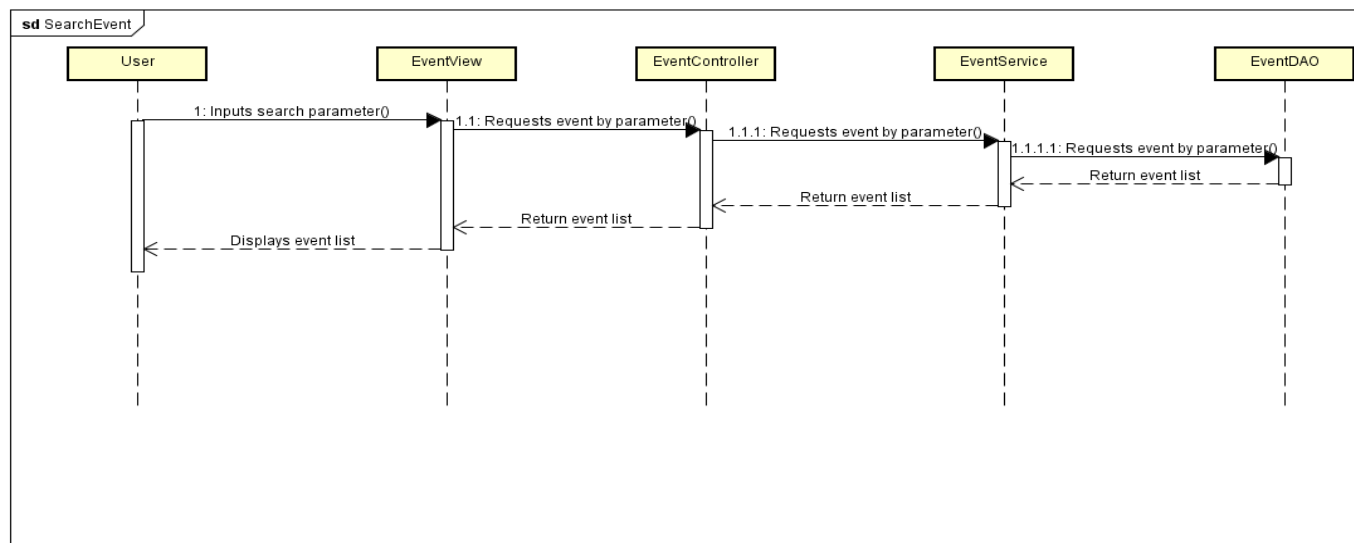
Legenda: O diagrama mostra as etapas de processo e lógica percorridas para um cadastro de eventos. Fonte: Os autores.

Os diagramas para edição e exclusão de um cadastro percorrem as mesmas etapas dos diagramas de inclusão, apenas com ações diferentes do usuário, que clica em um botão de “EDITAR” ou de “EXCLUIR”, acionando uma query diferente no banco de dados.

6.9 APÊNDICE I – DIAGRAMA DE SEQUÊNCIA DE BUSCA POR EVENTO

A Figura 38 apresenta um diagrama com as camadas percorridas no processo de busca por um evento dentro do sistema. Eventos podem ser buscados pelo usuário através de palavras-chave associadas a estes eventos. As palavras-chave são enviadas como parâmetros até o banco de dados, que retorna o resultado da *request*.

FIGURA 38 – DIAGRAMA DE SEQUÊNCIA DE BUSCA POR EVENTO

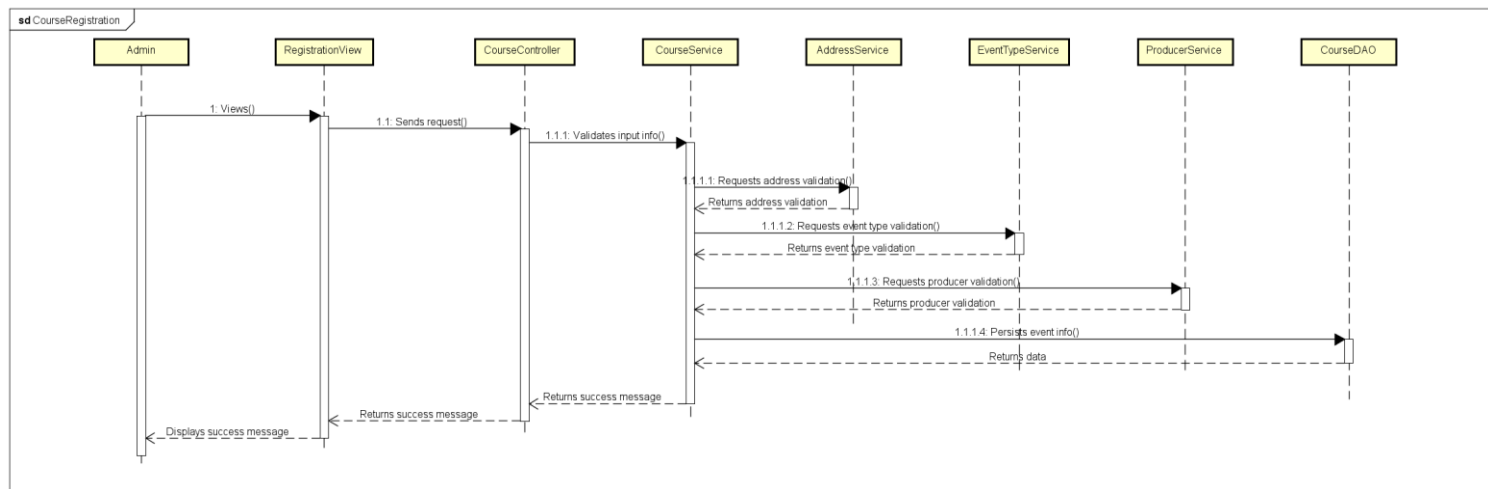


Legenda: a Figura acima apresenta o diagrama com o fluxo de busca por um evento dentro do sistema. Fonte: Os autores.

6.10 APÊNDICE J – DIAGRAMA DE SEQUÊNCIA DE CADASTRO DE CURSO

Muito semelhante ao cadastro de eventos, o que diferencia o processo no cadastro de cursos são o controller e o service acessados para envio e validação da *request*.

FIGURA 39 – DIAGRAMA DE SEQUÊNCIA PARA CADASTRO DE CURSOS



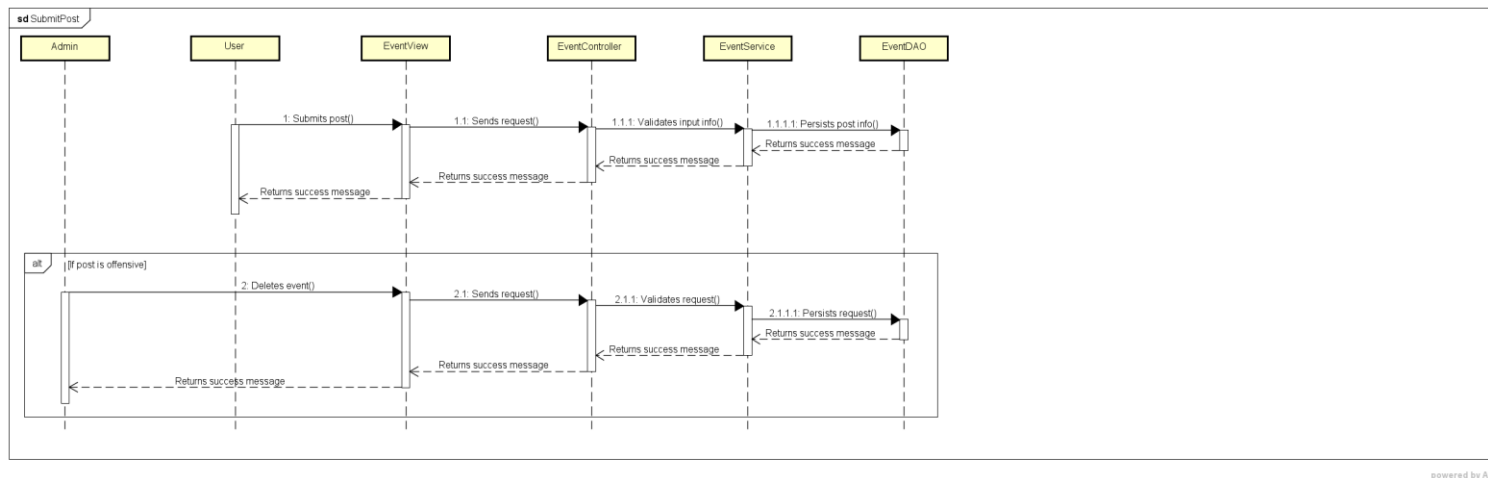
powered by Astah

Fonte: Os autores.

6.11 APÊNDICE K – DIAGRAMA DE SEQUÊNCIA PARA POSTAGEM EM EVENTOS

O usuário tem a opção de realizar postagem de comentários em eventos, como mostra a Figura 40.

FIGURA 40 – DIAGRAMA DE SEQUENCIA DE REALIZAÇÃO DE POSTAGEM

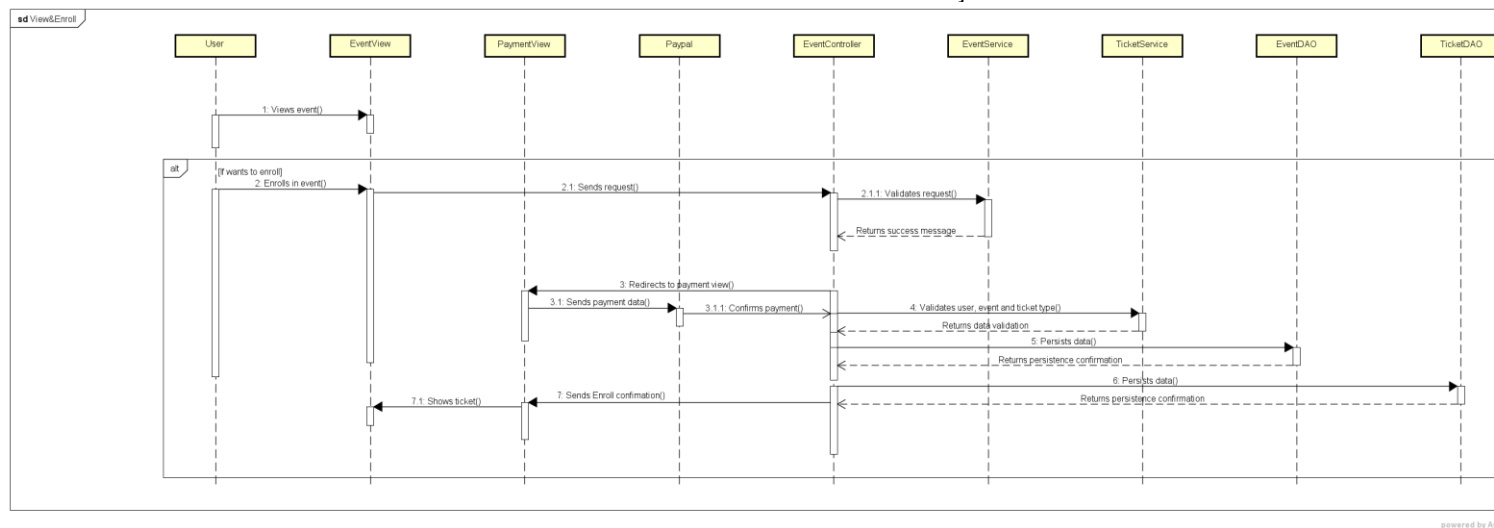


Legenda: a Figura acima mostra o processo percorrido para que a postagem de um usuário seja efetiva. Alternativamente, há o caminho percorrido caso um administrador decida apagar uma postagem. Fonte: Os autores.

6.12 APÊNDICE L – DIAGRAMA DE SEQUÊNCIA PARA INSCRIÇÃO EM EVENTO

Todos os usuários podem visualizar os eventos no *app*, mas precisam ter seus dados validados e realizarem um pagamento através do paypal para que possam se inscrever em um evento e receber um ticket para o mesmo, como é apresentado na Figura 41.

FIGURA 41 – DIAGRAMA DE SEQUÊNCIA PARA INSCRIÇÃO EM EVENTO



Fonte: Os autores.

Apêndice M – Descrição dos testes

Requisito 1 (R1) – Formulários incompletos

O formulário deve apresentar mensagem de erro quando o botão salvar for pressionado sem que todos os campos obrigatórios tenham sido preenchidos.

- Classes de equivalência válidas:
 - Mensagem de erro caso haja algum campo não preenchido;
 - Formulário não submetido.
- Classes de equivalência inválidas:
 - Mensagem de erro não apresentada ao tentar enviar formulário sem preencher todos os campos obrigatórios;
 - Formulário submetido sem preencher todos os campos obrigatórios;

Teste 1 (R1T1)

Nos formulários de eventos, cursos e ingressos o botão de envio não fora habilitado enquanto todos os campos obrigatórios não foram preenchidos.

Teste 2 (R1T2)

Nos demais formulários, o botão permanece habilitado o tempo todo, mas uma mensagem de erro é apresentada ao tentar finalizar o formulário sem todas as entradas obrigatórias e o formulário não é submetido.

Requisito 2 (R2) – CPF inválido

O formulário deve apresentar mensagem de erro ao tentar salvá-lo com um número de CPF inválido.

- Classes de equivalência válidas:
 - CPF com 11 dígitos numéricos;
 - Dígito verificador válido.
- Classes de equivalência inválidas:
 - CPF com mais de 11 dígitos;
 - CPF com menos de 11 dígitos;
 - Dígito verificador inválido.

Teste 1 (R2T1)

Os formulários de administradores e usuários inserem máscara à entrada e não permitem que seja informado número superior a 11 dígitos, apenas numéricos, nem permitem que seja submetido com menos de 11 dígitos.

Requisito 3 (R3) – Campos numéricos

Os campos que recebem valores especificamente numéricos não devem aceitar qualquer outro tipo de caractere.

- Classes de equivalência válidas:
 - Campos numéricos permanecem em branco sob o dígito de qualquer outro tipo de caractere.
- Classes de equivalência inválidas:
 - Campos numéricos aceita caracteres não-numéricos;
 - Formulários são submetidos com dados inválidos sem mensagem de erro.

Teste 1 (R3T1)

Nenhum campo numérico aceita caracteres não-numéricos. Os únicos caracteres especiais apresentados nesses campos são os das máscaras inseridas automaticamente.

Requisito 4 (R4) – Buscas

Deve ser possível buscar um ou mais itens através dos dados de qualquer coluna apresentada pela interface do sistema. As buscas devem funcionar com palavras inteiras ou apenas com parte delas.

- Classes de equivalência válidas:
 - Campos numéricos permanecem em branco sob o dígito de qualquer outro tipo de caractere.
- Classes de equivalência inválidas:
 - Campos numéricos aceita caracteres não-numéricos;
 - Formulários são submetidos com dados inválidos sem mensagem de erro.

Teste 1 (R4T1)

Em qualquer lista é possível buscar um ou mais itens através de palavras-chave que são comparadas com todos os dados apresentados nas colunas e trazem dinamicamente todos os resultados que coincidam.

Requisito 5 (R5) – Edição

Ao editar qualquer campo de um item criado inserindo dados válidos, deve ser possível salvá-lo novamente com os dados novos.

- Classes de equivalência válidas:
 - Campos editados com dados válidos podem ser salvos;

- Se cancelado antes de salvar, os dados antigos devem permanecer.
- Classes de equivalência inválidas:
 - Dados inválidos podem ser inseridos e salvos;
 - Item não pode ser salvo novamente;
 - Campos editados, após um cancelamento, perdem os dados.

Teste 1 (R5T1)

Todos os formulários de edição salvam dados válidos normalmente.

Teste 2 (R5T2)

Todos os formulários não permitem edição com dados inválidos. Nos formulários de cursos, eventos e ingressos, não é habilitado o botão de salvar a edição. Nos demais, é apresentada uma mensagem de erro.

Teste 3 (R5T3)

Após o cancelamento, nenhum editado é perdido ou alterado.

Requisito 6 (R6) – Exclusão

Ao excluir qualquer item criado, este deve sumir da lista de itens.

- Classes de equivalência válidas:
 - Item excluído some da lista de itens.
- Classes de equivalência inválidas:

Item excluído permanece na lista de itens.

